

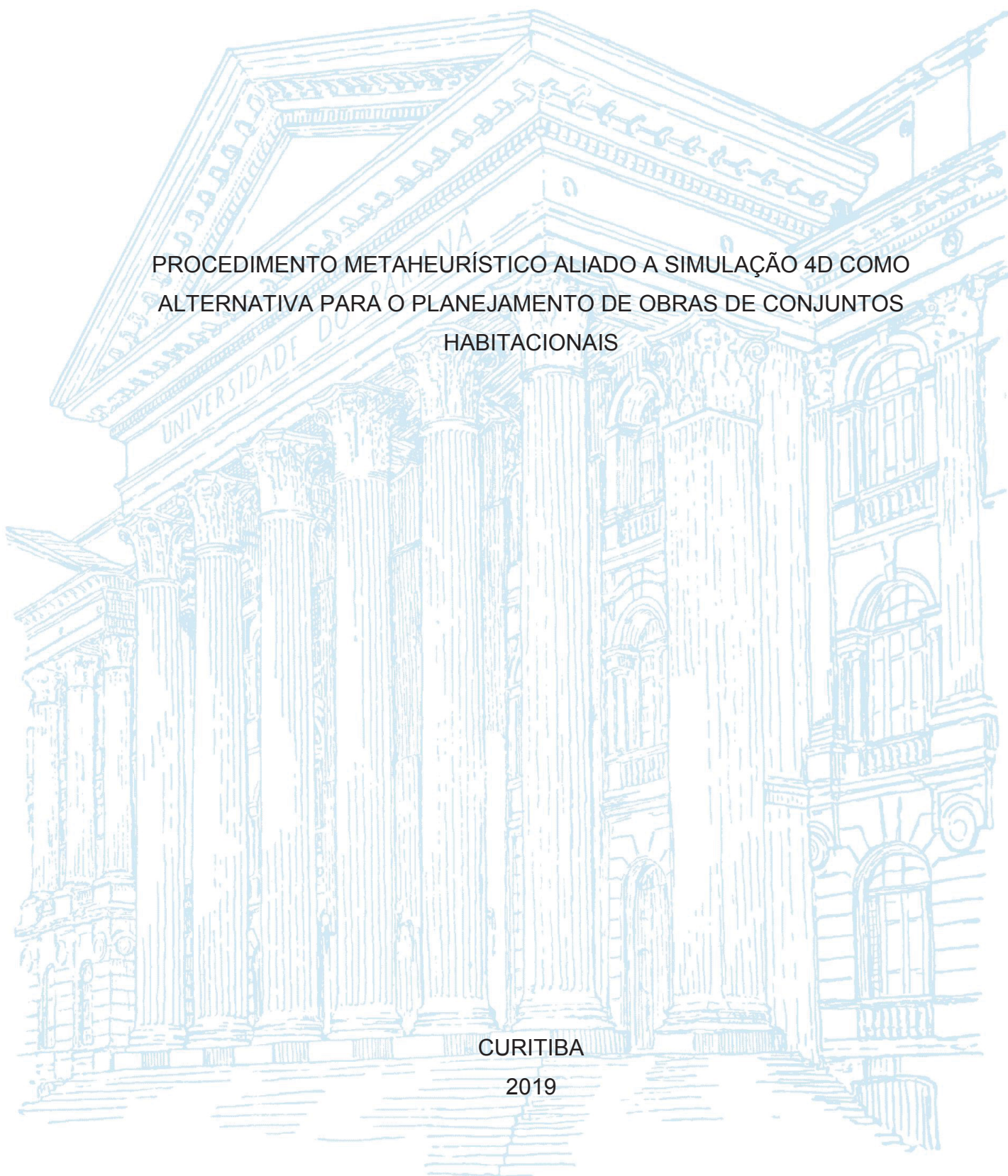
UNIVERSIDADE FEDERAL DO PARANÁ

PEDRO HENRIQUE PINTO BEZERRA

PROCEDIMENTO METAHEURÍSTICO ALIADO A SIMULAÇÃO 4D COMO  
ALTERNATIVA PARA O PLANEJAMENTO DE OBRAS DE CONJUNTOS  
HABITACIONAIS

CURITIBA

2019



PEDRO HENRIQUE PINTO BEZERRA

PROCEDIMENTO METAHEURÍSTICO ALIADO A SIMULAÇÃO 4D COMO  
ALTERNATIVA PARA O PLANEJAMENTO DE OBRAS DE CONJUNTOS  
HABITACIONAIS

Dissertação apresentada ao curso de Pós-Graduação em Engenharia de Construção Civil, Setor de Tecnologia, Universidade Federal do Paraná, como requisito para a obtenção do título de mestre em Engenharia Civil.

Orientador: Prof. Dr. Sergio Scheer

CURITIBA

2019

Catálogo na Fonte: Sistema de Bibliotecas, UFPR  
Biblioteca de Ciência e Tecnologia

B574p

Bezerra, Pedro Henrique Pinto

Procedimento metaheurístico aliado a simulação 4D como alternativa para o planejamento de obras de conjuntos habitacionais / Pedro Henrique Pinto Bezerra. – Curitiba, 2019.

Dissertação - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia de Construção Civil, 2019.

Orientador: Sergio Scheer.

1. Pesquisa operacional. 2. Algoritmos genéticos. 3. Conjuntos habitacionais. I. Universidade Federal do Paraná. II. Scheer, Sergio. III. Título.

CDD: 728.31

Bibliotecária: Vanusa Maciel CRB- 9/1928



MINISTÉRIO DA EDUCAÇÃO  
SETOR SETOR DE TECNOLOGIA  
UNIVERSIDADE FEDERAL DO PARANÁ  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO ENGENHARIA DE  
CONSTRUÇÃO CIVIL - 40001016049P2

## TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA DE CONSTRUÇÃO CIVIL da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **PEDRO HENRIQUE PINTO BEZERRA** intitulada: **Procedimento metaheurístico aliado a simulação 4d como alternativa para o planejamento de obras de conjuntos habitacionais**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 23 de Abril de 2019.

SÉRGIO SCHEER

Presidente da Banca Examinadora (UFPR)

PAULO HENRIQUE SIQUEIRA

Avaliador Externo (UFPR)

RICARDO MENDES JUNIOR

Avaliador Externo (UFPR)

ADRIANA DE PAULA LACERDA SANTOS

Avaliador Interno (UFPR)

Aos meus pais, Pedro e Dilma,  
por todo o suporte e incentivo  
durante esta caminhada.

## **AGRADECIMENTOS**

Ao prof. Dr. Sergio Scheer, pela orientação que permitiu a melhor realização deste trabalho.

Aos professores do Programa de Pós-Graduação em Engenharia de Construção Civil da UFPR, pelos conhecimentos partilhados durante meu período de mestrado.

À minha irmã, Adriana, e meu cunhado, Luiz, por serem alunos e profissionais exemplares que sempre me inspiraram a fazer o meu melhor.

À Valéria, Julianna e Drielle, por terem me recebido tão bem em uma nova cidade e por terem sido as melhores colegas de trabalho possíveis durante os últimos dois anos.

À Lívia, por ter estado do meu lado desde o início desta trajetória.

Aos amigos Alice, Júlia, Bruna, Denise, Lara, Luiz e Renato, por serem, hoje em dia, uma verdadeira família que Curitiba me presenteou.

Aos amigos Raif, Ianie e Amanda, que me acompanharam desde o início da graduação, e que mesmo não estando mais tão presentes do meu dia-a-dia, representam uma parte muito especial da minha trajetória.

## RESUMO

O presente trabalho tem como foco a aplicação de um procedimento meta-heurístico, representado pelo Algoritmo Genético, na busca de melhores alternativas de organização das atividades de obras de conjuntos habitacionais. Desse modo, o algoritmo é implementado de forma a buscar soluções para o Problema de Programação de Projetos com Restrições de Recursos (*Resource-Constrained Project Scheduling Problem – RCPSP*). A modelagem do problema é adaptada ao tipo de projeto em questão, onde é realizada a definição das relações entre as atividades de construção de uma unidade habitacional, por meio de um gráfico no tipo atividade-no-nó, e em seguida é utilizado o método da Linha de Balanceamento para determinar as datas de início e fim para todas as outras atividades repetitivas do empreendimento. Duas variações do Algoritmo Genético são implementadas, uma utilizando a técnica de seleção por Elitismo (GA) e outra utilizando uma técnica de hibridização com o algoritmo *Simulated Annealing* (GASA). Como forma de testar a eficiência dos algoritmos, são realizados ainda numerosos experimentos com instâncias da PSPLIB, uma base de dados para *benchmark* específica para o RCPSP, disponível online. Por fim, como forma de visualizar os cronogramas obtidos na fase de otimização, é realizada a simulação 4D, no *software Autodesk Navisworks*, do cronograma de um projeto de conjunto habitacional, gerado por meio do procedimento meta-heurístico implementado. A fase de simulação permite visualizar a construtibilidade do projeto, confirmar se o cronograma gerado segue uma boa lógica construtiva, assim como dá suporte à tomada de decisão sobre as melhores formas de execução das obras.

Palavras-chave: Pesquisa Operacional. Planejamento. Tomada de decisão. Meta-heurística. Simulação 4D.



## **ABSTRACT**

The present work focuses on the application of a metaheuristic procedure, represented by a Genetic Algorithm, in the search of better alternatives for the organization of activities in the construction of housing complexes. Therefore, the algorithm is implemented in order to solve the Resource-Constrained Project Scheduling Problem – RCPSP. The problem modeling is adapted to the type of project in question, where, at first, the activity network of a single housing unit is defined through an activity-on-the-node graph and afterwards the Line of Balance method is used in order to determine the start and finish dates of all the other repetitive activities of the project. Two variations of the Genetic Algorithm are implemented. The first, called GA, uses the Elitism selection technique, while the other, called GASA, uses a hybrid approach with the Simulated Annealing algorithm. In order to test the efficiency of the algorithms, numerous experiments are performed with project instances from the PSPLIB, a specific benchmark database for the RCPSP, available online. Finally, in order to better visualize the schedules obtained in the optimization phase, a 4D simulation of a schedule for the execution of a housing complex, generated with the metaheuristic procedure, is performed with the software Autodesk Navisworks. The simulation phase allows the visualization of the projects constructability, makes it possible to confirm if the algorithm-generated schedule follow a logical sequence, as well as it supports the decision-making about the best execution alternatives.

Key-words: Operations Research. Planning. Decision-Making. Metaheuristics. 4D Simulation.



## SUMÁRIO

|  |           |
|--|-----------|
| <b>1. INTRODUÇÃO .....</b>   | <b>14</b> |
| 1.1 DEFINIÇÃO DO PROBLEMA DE PESQUISA .....  | 15        |
| 1.2 OBJETIVOS .....  | 16        |
| 1.2.1 Objetivo Geral .....   | 16        |
| 1.2.2 Objetivos Específicos .....  | 17        |
| 1.3. LIMITAÇÕES DO TRABALHO .....  | 17        |
| 1.4 JUSTIFICATIVA .....  | 18        |
| <b>2. REFERENCIAL TEÓRICO.....</b>   | <b>21</b> |
| 2.1 A NECESSIDADE DE NOVAS ABORDAGENS DE PLANEJAMENTO PARA HABITAÇÃO DE INTERESSE SOCIAL ..... | 21        |
| 2.2 APRIMORAMENTO NO PROCESSO DE TOMADA DE DECISÃO: CONCEITOS DE PESQUISA OPERACIONAL .....    | 23        |
| 2.3 O PROCESSO DE PLANEJAMENTO DE PROJETOS LINEARES NA CONSTRUÇÃO CIVIL .....                  | 27        |
| 2.3 O PROBLEMA DA PROGRAMAÇÃO DE PROJETOS COM RESTRIÇÕES DE RECURSOS .....                     | 31        |
| 2.4.1 Heurísticas de Programação Baseadas em Regras de Prioridade .....                        | 33        |
| 2.4.1.1 O Método Serial .....  | 34        |
| 2.4.1.2 O Método Paralelo .....  | 36        |
| 2.4.2. Computação Inspirada na Físico-química .....  | 37        |
| 2.4.3. Computação Evolucionária.....   | 41        |
| 2.5 ALGORITMOS GENÉTICOS .....   | 43        |
| 2.5.1 Operadores de Cruzamento ou Recombinação .....   | 46        |
| 2.5.2 Operadores de Mutação .....  | 48        |
| 2.5.3 Processos de Seleção.....  | 49        |
| 2.6 SIMULAÇÃO BIM 4D .....   | 50        |
| <b>3. MÉTODO.....</b>  | <b>54</b> |
| 3.1 ESTRATÉGIA DE PESQUISA .....   | 54        |
| 3.2. MODELAGEM DO PROBLEMA.....  | 56        |
| 3.2.1 Pressupostos para Programação .....  | 56        |
| 3.2.2 Características do Projeto .....   | 57        |
| 3.3 MÓDULO DE OTIMIZAÇÃO .....   | 60        |

|           |   |            |
|-----------|---|------------|
| 3.3.1     | Codificação das Soluções e Geração da População Inicial | 62         |
| 3.3.2     | Mecanismo de Avaliação de Aptidão                       | 63         |
| 3.3.3     | Operadores de Seleção                                   | 66         |
| 3.3.4     | Operador de Cruzamento                                  | 68         |
| 3.3.5     | Operador de Mutação                                     | 68         |
| 3.3.6     | Critério de Parada                                      | 69         |
| 3.4       | MÓDULO DE SIMULAÇÃO                                     | 69         |
| <b>4.</b> | <b>RESULTADOS</b>                                       | <b>70</b>  |
| 4.1       | EXPERIMENTOS COMPUTACIONAIS                             | 70         |
| 4.2       | EXEMPLO DE APLICAÇÃO DO MÓDULO DE SIMULAÇÃO             | 77         |
| 4.3       | ANÁLISE DOS RESULTADOS                                  | 85         |
| <b>5.</b> | <b>CONSIDERAÇÕES FINAIS</b>                             | <b>87</b>  |
| 5.1       | TRABALHOS FUTUROS                                       | 91         |
|           | <b>REFERÊNCIAS</b>                                      | <b>93</b>  |
|           | <b>APÊNDICE A</b>                                       | <b>99</b>  |
|           | <b>APÊNDICE B</b>                                       | <b>121</b> |
|           | <b>APÊNDICE C</b>                                       | <b>123</b> |
|           | <b>APÊNDICE D</b>                                       | <b>126</b> |

## LISTA DE FIGURAS

|  |    |
|--|----|
| FIGURA 1 – PROCESSO DE MODELAGEM.....  | 25 |
| FIGURA 2 – DIAGRAMA DA LINHA DE BALANCEAMENTO.....   | 30 |
| FIGURA 3 – ESTRUTURA EM PSEUDOCÓDIGO DE HEURÍSTICA DE<br>PROGRAMAÇÃO BASEADA EM REGRA DE PRIORIDADE.....   | 34 |
| FIGURA 4 – PSEUDOCÓDIGO DO ESQUEMA DE PROGRAMAÇÃO SERIAL..   | 35 |
| FIGURA 5 – PSEUDOCÓDIGO DO ESQUEMA DE PROGRAMAÇÃO<br>PARALELO.....   | 37 |
| FIGURA 6 – CONTROLE DE ACEITAÇÃO DE NOVAS SOLUÇÕES.....  | 40 |
| FIGURA 7 – ESTRUTURA BÁSICA EM PSEUDOCÓDIGO DO <i>SIMULATED<br/>ANNEALING</i> .....  | 40 |
| FIGURA 8 – ESTRUTURA DE UM CROMOSSOMO.....   | 42 |
| FIGURA 9 – ESTRUTURA BÁSICA DOS ALGORITMOS GENÉTICOS.....  | 43 |
| FIGURA 10 – ESTRUTURA BÁSICA EM PSEUDOCÓDIGO DE UM<br>ALGORITMO GENÉTICO.....  | 45 |
| FIGURA 11 – OPERADOR DE UM PONTO.....  | 47 |
| FIGURA 12 – OPERADOR DE DOIS PONTOS.....   | 47 |
| FIGURA 13 – OPERADOR UNIFORME.....   | 50 |
| FIGURA 14 – OPERADORES DE MUTAÇÃO.....   | 50 |
| FIGURA 15 – SELEÇÃO POR ROLETA.....  | 50 |
| FIGURA 16 – INTER-RELAÇÕES ENTRE ATIVIDADES E MÉTODO DE<br>PROGRAMAÇÃO DE PROJETOS REPETITIVOS.....  | 57 |
| FIGURA 17 – REPRESENTAÇÃO DE UMA UNIDADE HABITACIONAL DO<br>PROJETO EM ATIVIDADE-NO-NÓ E REPRESENTAÇÃO DA REPETIÇÃO<br>POR MEIO DO MÉTODO DE LINHA DE BALANCEAMENTO..... | 58 |
| FIGURA 18 – FLUXOGRAMA DE OPERAÇÃO DO ALGORITMO PROPOSTO.  | 60 |
| FIGURA 19 – FLUXOGRAMA DA HIBRIDIZAÇÃO COM <i>SIMULATED<br/>ANNEALING</i> (GASA).....  | 61 |
| FIGURA 20 – SEQUENCIAMENTOS VIÁVEIS E INVIÁVEIS DE ACORDO<br>COM AS RELAÇÕES DE PRECEDÊNCIA.....   | 63 |
| FIGURA 21 – ANÁLISE DA UNIDADE CRÍTICA PARA O CASO 1.....  | 65 |
| FIGURA 22 – ANÁLISE DA UNIDADE CRÍTICA PARA O CASO 2.....  | 66 |

|   |    |
|---|----|
| FIGURA 23 – PROJETO DE EDIFÍCIO PADRÃO MICHA CASA MINHA<br>VIDA.....  | 78 |
| FIGURA 24 – SIMULAÇÃO DO PROJETO DE EDIFÍCIO NO<br><i>NAVISWORKS</i> .....  | 79 |
| FIGURA 25 – MODELO DO PROJETO DE CONJUNTO HABITACIONAL COM<br>100 UNIDADES.....   | 81 |
| FIGURA 26 – GRÁFICO ATIVIDADE-NO-NÓ DAS RELAÇÕES ENTRE AS<br>ATIVIDADES PARA A EXECUÇÃO DE UMA UNIDADE<br>HABITACIONAL..... | 81 |
| FIGURA 27 – ARQUIVO .CSV GERADO PELO PROGRAMA.....  | 82 |
| FIGURA 28 – IMPORTAÇÃO PARA O <i>NAVISWORKS</i> .....   | 82 |
| FIGURA 29 – GRÁFICO DE GANTT GERADO PELO <i>NAVISWORKS</i> .....  | 83 |
| FIGURA 30 – SIMULAÇÃO DO PROJETO DE CONJUNTO HABITACIONAL<br>NO <i>NAVISWORKS</i> .....                                     | 84 |
| FIGURA 31 – LINHA DE BALANCEAMENTO PARA O PROJETO SIMULADO..  | 84 |

## LISTA DE TABELAS

|   |    |
|---|----|
| TABELA 1 – TERMOS MAIS UTILIZADOS EM COMPUTAÇÃO<br>EVOLUCIONÁRIA.....             | 42 |
| TABELA 2 – DURAÇÃO E RELAÇÕES DE PRECEDÊNCIA DO PRIMEIRO<br>PROJETO SIMULADO..... | 77 |
| TABELA 3 – DURAÇÃO E RELAÇÕES DE PRECEDÊNCIA DO PROJETO<br>CONSIDERADO.....       | 80 |

## LISTA DE GRÁFICOS

|   |    |
|---|----|
| GRÁFICO 1 – NÚMERO DE PUBLICAÇÕES EM QUE CADA TÉCNICA FOI UTILIZADA.....  | 15 |
| GRÁFICO 2 – DIFERENÇA PERCENTUAL ENTRE A MELHOR SOLUÇÃO ENCONTRADA E A SOLUÇÃO ÓTIMA/LOWER BOUND POR INSTÂNCIA..... | 71 |
| GRÁFICO 3 – DIFERENÇA PERCENTUAL ENTRE PIOR SOLUÇÃO ENCONTRADA E A SOLUÇÃO ÓTIMA/LOWER BOUND POR INSTÂNCIA.....     | 62 |
| GRÁFICO 4 – TEMPO MÉDIO PARA ENCOTRAR AS SOLUÇÕES.....  | 73 |
| GRÁFICO 5 – NÚMERO MÉDIO DE ITERAÇÕES DESEMPENHADAS PARA CADA INSTÂNCIA.....  | 73 |
| GRÁFICO 6 – COMPORTAMENTO DA MELHORIA DAS SOLUÇÕES AO LONGO DAS GERAÇÕES PARA O GA – 30 ATIVIDADES.....             | 74 |
| GRÁFICO 7 – COMPORTAMENTO DA MELHORIA DAS SOLUÇÕES AO LONGO DAS GERAÇÕES PARA O GASA – 30 ATIVIDADES.....           | 74 |
| GRÁFICO 8 – COMPORTAMENTO DA MELHORIA DAS SOLUÇÕES AO LONGO DAS GERAÇÕES PARA O GA – 60 ATIVIDADES.....             | 75 |
| GRÁFICO 9 – COMPORTAMENTO DA MELHORIA DAS SOLUÇÕES AO LONGO DAS GERAÇÕES PARA O GASA – 60 ATIVIDADES.....           | 75 |
| GRÁFICO 10 – COMPORTAMENTO DA MELHORIA DAS SOLUÇÕES AO LONGO DAS GERAÇÕES PARA O GA – 90 ATIVIDADES.....            | 76 |
| GRÁFICO 11 – COMPORTAMENTO DA MELHORIA DAS SOLUÇÕES AO LONGO DAS GERAÇÕES PARA O GASA – 90 ATIVIDADES.....          | 76 |

## 1. INTRODUÇÃO

O uso de técnicas computacionais para o aprimoramento do planejamento de obras já é abordado em pesquisas há várias décadas, sendo, hoje em dia, cada vez mais empregado em conjunto com *software* de concepção de projetos e de simulação de processos, como forma de subsidiar a tomada da decisão tanto em fases pré-operacionais, como durante o processo construtivo. Em Bezerra e Scheer (2018), observa-se uma predominância no uso de algoritmos meta-heurísticos e de simulação de eventos discretos em conjunto com modelos BIM, sendo possível identificar uma adoção cada vez maior de conceitos da área de pesquisa operacional no setor de construção. Estas técnicas, por sua vez, são capazes de tratar diversos objetivos específicos ao planejamento de obras, seja a organização de atividades, o gerenciamento de recursos e até mesmo a configuração dos canteiros de obras.

No contexto dos empreendimentos públicos, sobretudo nas habitações de interesse social, verifica-se um potencial ainda maior para a aplicação da pesquisa operacional, em razão da natureza repetitiva destes empreendimentos, similar a uma linha de produção industrial. Com o déficit habitacional presenciado atualmente em países menos desenvolvidos, alternativas que tornem o processo de projeto e construção de habitações populares mais rápido são muito necessárias (BEZERRA; SANTOS; SCHEER, 2018). Nesse sentido, a adoção cada vez maior de *software* BIM e o estudo por maneiras de integrar essa tecnologia a procedimentos de tomada de decisão oriundos da pesquisa operacional configura uma agenda de pesquisa bastante atual.

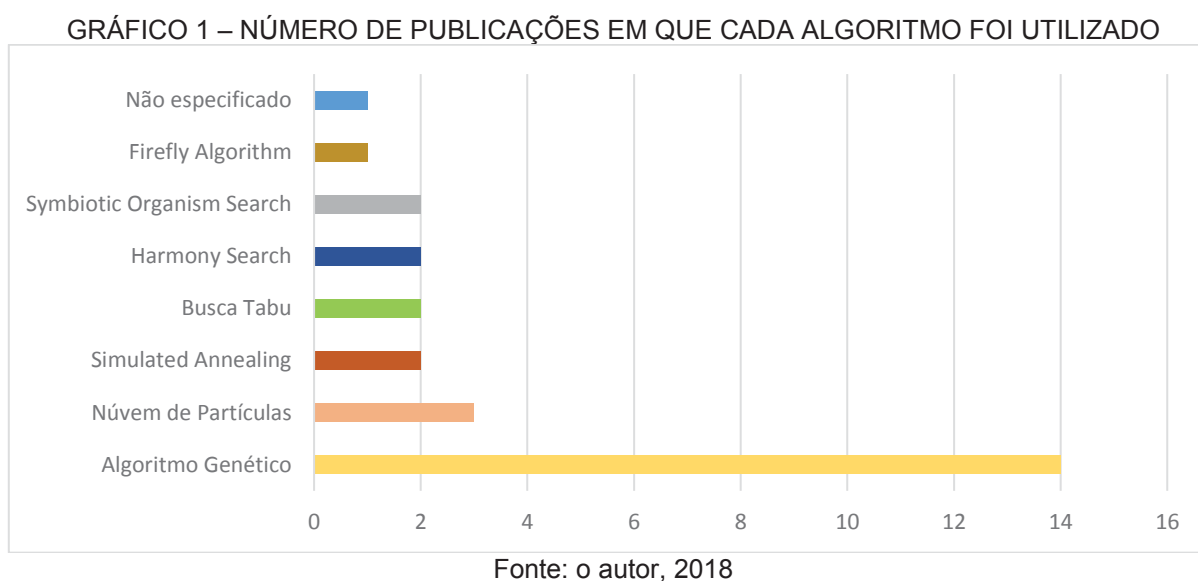
Na área de pesquisa operacional, no entanto, existem inúmeras técnicas utilizadas na melhoria das estratégias de execução de projetos. Assim, para a formulação do problema de pesquisa, é preciso primeiramente analisar quais são as principais técnicas utilizadas na área de construção civil, e em que tipos de processos elas são aplicadas, para que seja possível identificar a melhor ferramenta a ser utilizada no contexto da habitação de interesse social. Para isso, decidiu-se primeiramente pela condução de uma Revisão Sistemática da Literatura (ANEXO I).



## 1.1 DEFINIÇÃO DO PROBLEMA DE PESQUISA

A revisão sistemática realizada permitiu identificar diversas abordagens em que técnicas de pesquisa operacional foram utilizadas no contexto da construção civil, de modo a melhorar a tomada de decisão. Houve predominância, no entanto, do uso de algoritmos meta-heurísticos. Foi possível observar sua aplicação em publicações lidando com todos os quatro tipos de processos identificados.

Observou-se ainda o uso de diversos tipos de algoritmos nas pesquisas, como mostra o Gráfico 7.



Os estudos analisados evidenciaram maior relevância e conhecimento sobre a aplicação de algoritmos genéticos no contexto do planejamento de obras de construção. É possível constatar o grande potencial desta técnica em obter bons resultados para a programação de atividades no tempo (aprimoramento da duração) e no processo de alocação de recursos, sendo aplicado com sucesso em diversos estudos sobre a busca por soluções de problemas de otimização como o Problema de Alocação de Recursos (*Resource Leveling Problem* – RLP) e o Problema do Programação de Projetos com Restrições de Recursos (*Resource-Constrained Project Scheduling Problem* – RCPSP).

Segundo Marzouk et al. (2014), os projetos de habitação social, em geral, têm caráter repetitivo, ou seja, são compostos por um grande número de atividades que são realizadas várias vezes durante a execução. Métodos tradicionais de planejamento como CPM (*Critical Path Method*) e gráficos de Gantt, por sua vez, não são eficazes para planejar projetos com tais características. Mesmo depois do

surgimento de métodos específicos para empreendimentos repetitivos, como a Linha de Balanceamento, existe ainda a dificuldade de considerar incertezas no processo executivo e a interação entre os recursos envolvidos. Desse modo, devido a sua natureza repetitiva, e também por serem geralmente grandiosos, os projetos de habitação social demandam técnicas de gerenciamento mais complexas.

A presente dissertação busca investigar, então, novas abordagens para a tomada de decisão no mercado de construção que auxiliem no planejamento de conjuntos habitacionais de interesse social. Novas formas de planejar podem ser oportunas para este tipo de empreendimento no sentido de permitir a construção mais rápida e racionalizada, além de conscientizar os agentes envolvidos sobre a importância das fases pré-operacionais sobre o ciclo de vida do projeto. Essas novas abordagens poderiam ainda abrir caminho para o desenvolvimento dos programas de habitação não só quanto à quantidade de habitações construídas, como também quanto ao aumento da qualidade e menor padronização destas residências.

Assim, visto a necessidade de aprimorar os processos de planejamento vigentes para projetos de habitação de interesse social, buscar diminuir o déficit habitacional visto no Brasil e entregar habitações com maior qualidade à população, o presente trabalho busca responder ao seguinte questionamento: “De que maneira o uso de um algoritmo genético poderia ser aplicado ao planejamento da construção de conjuntos habitacionais de interesse social, buscando o aprimoramento do tempo de execução e o atendimento a restrições de recursos?”.

## 1.2 OBJETIVOS

Com a realização da Revisão Sistemática da Literatura, foi possível formular o problema de pesquisa, a partir do qual são a seguir definidos os objetivos geral e específicos da presente dissertação.

### 1.2.1 Objetivo Geral

O objetivo deste trabalho é investigar a eficiência da aplicação de técnicas computacionais como forma de subsidiar a tomada de decisão na fase de planejamento de obras de conjuntos habitacionais, por meio da implementação de um algoritmo genético aliado à simulação em *software* BIM 4D.

### 1.2.2 Objetivos Específicos

- Discutir as principais características do Problema de Programação de Projetos com Restrições de Recursos (*Resource-Constrained Project Scheduling Problem – RCPSP*) e o potencial da técnica de algoritmo genético para solucioná-lo;
- Implementar um algoritmo genético que forneça cronogramas aprimorados, com tempo de processamento e consumo de memória computacional razoáveis;
- Incluir na modelagem as características mais condizentes com situações reais, levando em consideração particularidades dos projetos de conjuntos habitacionais, de forma que o algoritmo possa ser aplicado na prática;
- Validar a eficiência do algoritmo nas bases de *benchmark* disponíveis na literatura;
- Aplicar o algoritmo a um projeto de habitação popular real, obtendo um cronograma de execução aprimorado; e
- Modelar o projeto em *software* BIM e realizar simulações 4D do cronograma fornecido pelo algoritmo, a fim de avaliar sua construtibilidade.

### 1.3. LIMITAÇÕES DO TRABALHO

As limitações do presente trabalho relacionam-se aos pressupostos usados para a programação da função de aptidão do Problema de Programação de Projetos com Restrições de Recursos e à codificação usada para implementação do algoritmo genético. São elas:

- Admite-se que duração de cada atividade, as relações de precedência, a quantidade de recursos utilizados e as limitações de recursos são previamente conhecidos, servindo como dados de entrada para o algoritmo genético;
- O ritmo de trabalho de cada tarefa é considerado constante durante sua execução em todas as unidades habitacionais, sem que haja interrupções;
- O processo descrito no Módulo de Simulação depende da qualidade da modelagem do projeto em *software* BIM;

- O único tipo de recurso considerado é o número de equipes usadas em cada tarefa. No entanto, podem ser considerados tipos de equipes diferentes trabalhando na mesma tarefa;
- Admite-se apenas um modo de execução para cada atividade;
- Os únicos fatores a serem aprimorados no sistema proposto são a duração do projeto e a alocação de recursos; e
- Considera-se todas as atividades como sendo repetitivas e realizadas obrigatoriamente em todas as unidades habitacionais do projeto.

#### 1.4 JUSTIFICATIVA

O crescimento urbano gera uma sobrecarga na necessidade de infraestrutura, afetando o funcionamento das cidades e comprometendo a qualidade de vida da população. O problema habitacional e as inadequadas condições de moradia da população de baixa renda também são problemas gerados por tal fenômeno. Assim, tornam-se necessários investimentos nas cidades, na tentativa de diminuir a problemática ocasionada pelo crescente aumento da urbanização (RUBIN; BOLFE, 2014).

A população urbana mundial deverá crescer de 3,6 bilhões em 2011 para 6,3 bilhões em 2050, com 94% deste aumento ocorrendo nos países em desenvolvimento. Estima-se que 828 milhões de pessoas de países em desenvolvimento vivam em favelas e moradias precárias, enquanto o número aumentará para 1,4 bilhão até 2020. Assim, acomodar os mais pobres representa um desafio significativo para estes países durante o processo de crescimento urbano (GAN et al., 2017)

Por sua vez, em todo o Brasil, a necessidade de moradia aumenta cada vez mais, conforme indicado pela crescente porcentagem da população vivendo em favelas. Entre 2000 e 2010, a população das favelas cresceu 75%, enquanto a população total aumentou apenas 12,3%. Em 2008, de acordo com o IBGE, havia 15 milhões de lares considerados “inadequados”, ou seja, que não apresentam infraestrutura básica ou título de propriedade ou que se encontram superlotados (DONAGHY, 2013).

A partir de 2005, elevaram-se, no Brasil, os investimentos para o financiamento habitacional direcionado à população de baixa renda. Em 2007, foi

anunciada pelo Governo Federal a implantação do Programa de Aceleração do Crescimento (PAC), que traz investimentos em várias áreas como infraestrutura, setores de habitação e saneamento e urbanização de assentamentos precários. Dando continuidade a esses investimentos, no ano de 2009, o governo lançou o programa “Minha Casa, Minha Vida” (MCMV), com o objetivo de subsidiar a aquisição da casa própria para famílias com renda até R\$ 1.600,00 e facilitar as condições de acesso ao imóvel para famílias com renda até R\$ 5 mil. Empresas privadas e públicas aderiram a esta ação, representando um enorme desafio para o governo, construtores e escritórios de arquitetura (OLIVA; GRANJA, 2013; RUBIN; BOLFE, 2014).

Desde sua criação, o MCMV despertou grande interesse da comunidade acadêmica. No estudo realizado por Kowaltowski et al. (2018), foi possível encontrar 2477 publicações sobre a temática entre os anos de 2010 e 2016. O interesse também pode ser relacionado ao tamanho do programa, no qual mais de três milhões de residências já foram construídas.

Kowaltowski et al. (2018) identificam, no entanto, contradições quanto aos benefícios trazidos pelo programa, sobretudo no que se refere à melhoria da qualidade de vida das pessoas. Além disso, questões relativas ao projeto simplório e padronizado das habitações e problemas de restrição orçamentária também foram levantados. Desse modo, além da questão social, também se identifica a demanda por novas abordagens de planejamento e processos de tomada de decisão como forma de garantir a construção de moradias que realmente atendam às necessidades das famílias beneficiadas pelo programa.

Conforme verificado em Bezerra, Santos e Scheer (2018), Bezerra e Scheer (2018) e na revisão sistemática presente no Anexo I, o planejamento de obras pode ser subsidiado por uma grande gama de técnicas, mas ainda há poucos estudos voltados para sua aplicação na construção de habitações de interesse social. O uso de procedimentos meta-heurísticos, por sua vez, se mostrou relevante na maioria dos trabalhos que tratam do aprimoramento de fatores relacionados ao gerenciamento e controle de processos construtivos. A grande quantidade de referências sobre essas técnicas, sua relativa facilidade de implementação, os bons resultados encontrados na literatura, como também a falta de estudos brasileiros que as implementem no setor de construção, foram, assim, determinantes na sua escolha como foco da presente dissertação.

Foi possível verificar também uma grande quantidade de publicações que integravam algoritmos meta-heurísticos com modelos BIM. As formas de integração são também variadas, mas o aspecto mais destacado foi a capacidade que os *softwares* BIM oferecem de visualizar aquilo que se construirá e também de realizar modificações prévias à execução. Assim, o presente trabalho procura integrar essa capacidade de visualização com programações otimizadas por um procedimento meta-heurístico, por meio do uso de simulação 4D da construção.

Desse modo, espera-se criar uma ferramenta que possa ser utilizada na prática e que reduza o tempo e o esforço humano despendidos no processo de planejamento de obras. No contexto da habitação social, o método descrito neste trabalho possivelmente poderá abrir caminho para futuras pesquisas que incluam outros aspectos importantes para o planejamento dessas obras além da duração, como custo, conforto ambiental, menor padronização e aumento da qualidade das obras.

## 2. REFERENCIAL TEÓRICO

A presente dissertação tem como foco o aprimoramento do processo de tomada de decisão quanto à execução de projetos de habitação de interesse social, a partir de técnicas da área de pesquisa operacional. Assim, será primeiramente evidenciada a necessidade de pesquisar novas técnicas que permitam aprimorar os processos que permeiam a execução desses tipos de projeto atualmente. Em seguida, será apresentado um breve histórico sobre pesquisa operacional, citando as principais técnicas da área e seu potencial de aplicação na indústria de construção. Por fim, serão expostas as bases teóricas para a formulação do método proposto neste trabalho.

### 2.1 A NECESSIDADE DE NOVAS ABORDAGENS DE PLANEJAMENTO PARA HABITAÇÃO DE INTERESSE SOCIAL

Historicamente, no Brasil, muitas políticas públicas investiram no setor de construção como forma de desenvolver a infraestrutura e promover o crescimento econômico do país. Na recente história econômica brasileira, políticas importantes incluíram os decretos-lei no período do governo de Getúlio Vargas (1930-1945), o Sistema Financeiro de Habitação (SFH), o Banco Nacional de Habitação (BNH) e, mais recentemente, o programa Minha Casa, Minha Vida (MCMV). Todas essas políticas públicas foram promovidas a nível federal (CAMPOS; GUILHOTO, 2017).

O MCMV é o maior programa de habitação de interesse social da América Latina. A iniciativa executa dois tipos básicos de unidades habitacionais: uma casa de 35 m<sup>2</sup> e um apartamento de 42 m<sup>2</sup> com uma superfície interna de 37 m<sup>2</sup>. As residências têm que ser construídas cumprindo certas condições com relação ao tamanho, arquitetura, uso de materiais e dimensões aplicadas. O progresso das obras é monitorado pela Caixa Econômica Federal até que as unidades sejam vendidas e as famílias beneficiárias tenham se mudado (SOMERS; BAUD, 2013).

Estudos sobre habitação são importantes por seu potencial de influenciar políticas públicas e impactar positivamente futuros projetos. No caso das pesquisas realizadas no âmbito do programa MCMV, o foco é maior nas consequências sociais e urbanas dos empreendimentos. Assim, a existência de pesquisas sobre técnicas de aprimoramento do processo de projeto e execução das construções ainda é limitada,



sendo mudanças no formato destes processos muito raras (KOWALTOWSKI et al., 2018).

Esta situação é alavancada pelo fato de que o MCMV descreve requisitos mínimos para os *layouts* das unidades habitacionais, qualidade da construção e urbanização. As empresas de construção, consequentemente, atendem a esses requisitos para que possam entregar habitações dentro de um orçamento muito baixo, estabelecido pelas normas do programa. Além disso, as habitações precisam ser acessíveis financeiramente à população, fazendo com que os empreendimentos fiquem divididos entre a diminuição dos custos e a criatividade projetual. Na maioria das vezes, as habitações nem mesmo são entregues aos moradores completamente finalizadas, de modo que estes ainda precisem investir em atividades de construção adicionais (KOWALTOWSKI et al., 2018).

Para Oliva e Granja (2013), as restrições orçamentárias às quais os projetos de moradia popular geralmente estão submetidos acarretam na perda de qualidade da construção. Por outro lado, estas restrições poderiam incentivar a criatividade e inovação nas fases pré-operacionais, a fim de melhor atender às metas de custo. Limitações orçamentárias e percepção de valor pelo usuário deveriam servir como gatilho para induzir a criatividade no processo de projeto, como forma de buscar soluções inovadoras que tragam mais qualidade ao produto entregue, levando em consideração os interesses de todos os agentes envolvidos.

No MCMV, o orçamento detalhado é executado somente após a conclusão do processo de projeto, enquanto a abordagem ideal é a discussão e o estabelecimento de metas de custo nos estágios iniciais da concepção do empreendimento, que devem ser atualizados frequentemente durante a execução, a fim de fornecer feedback para ações futuras (OLIVA; GRANJA, 2013). Há também estudos que adotam novos métodos, levando em consideração os requisitos dos usuários, e que propõem a realocação dos custos de modo a permitir que os projetos de habitações populares vão além dos padrões mínimos (KOWALTOWSKI et al., 2018).

Para Kowaltowski et al. (2018), as metas essenciais de projetos de habitação de interesse social são que estes empreendimentos atendam às expectativas dos usuários e às necessidades coletivas por habitação nas cidades, por meio de sistemas viáveis dos pontos de vista econômico, social e ambiental.

Pesquisas futuras devem focar, então, em descobertas específicas que não só apoiem esses objetivos, mas também que ofereçam estratégias que ocasionem uma mudança real na qualidade de vida dos beneficiários.

Desse modo, é possível identificar a necessidade de se investir em planejamento, sobretudo do que diz respeito à tomada de decisão antes da execução. Segundo Somers e Baud (2013), no contexto da habitação social, o primeiro passo do processo de tomada de decisão deve ser definir e analisar o problema, levando em conta que estas questões são percebidas de maneira diferente por várias partes.

Portanto, os valores e critérios, os limites, restrições e objetivos das partes interessadas precisam ser explicitados. Tal formulação do problema requer vários tipos de conhecimento. A próxima fase é encontrar uma solução para o problema, identificando, projetando e rastreando opções alternativas para resolvê-lo. As consequências de curto e longo prazo de cada alternativa, bem como uma análise de riscos e custos são necessárias para selecionar as alternativas ideais. Para tanto, o conhecimento existente de todos os atores envolvidos sobre a questão é necessário (SOMERS; BAUD, 2013).

Problemas complexos como os de planejamento, produção e coordenação são examinados, formulados e solucionados por áreas da Engenharia, Ciência da Computação e Pesquisa Operacional (GOLDBARG; GOLDBARG; LUNA, 2016). Esta última, por sua vez, como observado na revisão presente no Anexo I, é relevante para o tratamento de problemas de planejamento, que envolvem a tomada de decisão entre vários cenários possíveis, influenciados por restrições. No planejamento de obras, essa atividade é constante, sendo importante estudar aplicações das técnicas de Pesquisa Operacional no setor de construção, de forma a diminuir a dificuldade e aumentar a precisão na definição da melhor forma de executar projetos.

## 2.2 APRIMORAMENTO NO PROCESSO DE TOMADA DE DECISÃO: CONCEITOS DE PESQUISA OPERACIONAL

Desde o advento da Revolução Industrial, o mundo presencia um crescimento extraordinário no tamanho e complexidade das organizações. Um fator crucial para essas mudanças foi o grande aumento na divisão do trabalho e a segmentação das responsabilidades gerenciais nessas empresas. Assim, à medida que aumentam a complexidade e a especialização em uma corporação, torna-se mais

difícil alocar os recursos disponíveis para as diversas atividades da maneira mais eficiente para a empresa como um todo. Esses tipos de problemas e a necessidade de encontrar o melhor caminho para solucioná-los criaram condições necessárias para o surgimento da *pesquisa operacional* (comumente chamada de **PO**) (HILLIER; LIEBERMAN, 2010).

Um dos principais fatores que impulsionaram o crescimento da área foi a revolução computacional. É preciso grande volume de processamento de cálculos para o tratamento eficiente de problemas complexos, tipicamente estudados pela PO. Assim, o desenvolvimento de computadores, com a capacidade de realizar cálculos muito mais rapidamente que o ser humano, deu grande impulso a este campo de estudo. O surgimento de computadores pessoais a partir dos anos 1980 também contribuiu para expansão da PO, que a partir de então ficou ao alcance de um número muito maior de pessoas (HILLIER; LIEBERMAN, 2010).

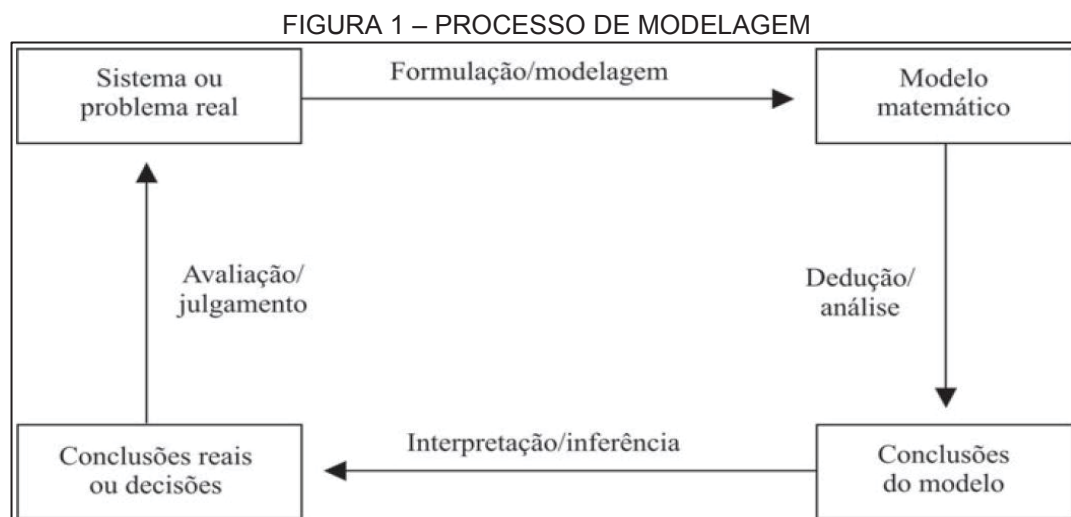
De forma resumida, a pesquisa operacional pode ser definida como o desenvolvimento de métodos científicos referentes a sistemas complexos, com a finalidade de prever e comparar estratégias ou decisões alternativas. O objetivo é dar suporte à definição de políticas e determinação de ações de forma científica. Para alguns autores, pesquisa operacional significa a abordagem científica para a tomada de decisões, que procura determinar como melhor projetar e operar um sistema, usualmente sob condições que requerem a alocação de recursos escassos. Assim, é possível definir, de maneira sucinta, que a PO é um enfoque científico e tecnológico sobre a tomada de decisões (ARENALES et al., 2011).

O componente científico está relacionado a ideias e processos para articular e modelar problemas de decisão, determinando os objetivos do tomador de decisão e as restrições sob as quais se deve operar. O componente tecnológico está relacionado a ferramentas de *software* e *hardware* para coletar, comunicar dados, e organizá-los. Estas informações são, então, utilizadas para gerar e aperfeiçoar modelos e reportar resultados. É possível afirmar até mesmo que a pesquisa operacional está se tornando cada vez mais um importante elemento da área de tecnologia da informação (ARENALES et al., 2011).

Os estudos de PO consistem basicamente na aplicação de um método científico na organização de atividades, de maneira a conduzir à resolução de problemas. O processo se inicia com a definição do problema, na qual devem ser

destacados os objetivos procurados, as relações entre as variáveis existentes, e as restrições que pesam sobre o sistema. A partir do conhecimento dessas características, é construída uma representação formal do estudo, que consiga expressar da maneira mais exata possível a sua forma de funcionamento. Esta representação, em geral, recebe o nome de *modelo*, que é na maioria das vezes expresso por meio de relações matemáticas (VON ALLENRIEDER, 1971).

Dispondo de um modelo, procura-se fazer um procedimento analítico que leve à resolução do problema. Assim, são realizados experimentos com as variáveis do sistema, a fim de alguma grandeza que expressa uma característica desejável ao processo analisado atinja um valor ótimo, ou suficientemente próximo do ótimo (VON ALLENRIEDER, 1971). O processo pode ser dividido em quatro etapas principais (Figura 1):



Fonte: Arenales et al. (2011)

Arenales et al. (2011) definem a abordagem de resolução de problemas por meio de pesquisa operacional como composta por várias fases baseadas da Figura 1: (i) definição do problema; (ii) construção do modelo; (iii) solução do modelo; (iv) validação do modelo; e (v) implementação da solução.

A fase (i) define o escopo do problema em estudo. A fase (ii) traduz a fase (i) em relações matemáticas ou lógicas de simulação. A fase (iii) utiliza métodos de solução e algoritmos conhecidos para resolver o modelo da fase (ii). A fase (iv) verifica se o modelo proposto representa apropriadamente o problema, ou seja, se o modelo prediz adequadamente o comportamento do sistema. Dentre os métodos de solução e algoritmos utilizados no processo, é possível citar: programação linear, programação

linear inteira, programação em redes, programação não-linear e meta-heurísticas (HILLIER; LIEBERMAN, 2010; ARENALES et al., 2011).

Por sua vez, no Anexo I são identificadas diversas dessas ferramentas em pesquisas na área de construção civil, sobretudo do que tange o planejamento de atividades e utilização de recursos na execução dos empreendimentos, como: programação linear, programação linear inteira e vários tipos de meta-heurísticas. Em Faghihi et al. (2015) e Marzouk et al. (2010), também é evidenciado um conjunto amplo de técnicas capazes de diminuir o esforço humano no processo de planejamento. Porém, embora os estudos tenham contribuído para a evolução destas técnicas para o aprimoramento de processos construtivos, não há muitas pesquisas que foquem em aplica-las no contexto da habitação de interesse social.

De acordo com os autores, ainda é, no entanto, necessário investir em pesquisa na área, devido à dependência que a atividade de programação tem não só da experiência dos profissionais, mas da própria estrutura analítica do projeto e dos recursos disponíveis. Uma das maiores lacunas existentes é a necessidade de uma abordagem capaz de avaliar múltiplos objetivos, aprimorando a organização e os custos, mas ao mesmo garantindo a estabilidade estrutural do projeto (FAFHIHI et al., 2015).

Para projetos de natureza repetitiva, como os de conjuntos habitacionais, também chamados de projetos lineares, a manutenção da continuidade do trabalho das equipes pode gerar melhorias em objetivos como o aumento da produtividade e redução nos custos com mão-de-obra, mesmo que isso signifique um aumento na duração do projeto. Profissionais encarregados do planejamento devem analisar constantemente a influência de diversos fatores sobre a condução do projeto, em busca de uma configuração equilibrada entre os diversos objetivos. Métodos tradicionais de planejamento de projetos repetitivos se mostram pouco eficientes para esses tipos de análise, uma vez que demandam muito trabalho e são capazes de gerar apenas uma solução para o empreendimento (HYARI; EL-RAYES, 2006). Nesse sentido, as técnicas de pesquisa operacional anteriormente citadas oferecem a oportunidade de promover melhorias no processo de tomada de decisão aplicado a estes empreendimentos, justamente por serem capazes de considerar diversas questões importantes por meio de sistemas automatizados.

## 2.3 O PROCESSO DE PLANEJAMENTO DE PROJETOS LINEARES NA CONSTRUÇÃO CIVIL

Segundo Ichihara (1999), na construção civil, denominam-se Projetos Lineares àqueles caracterizados por um número significativo de atividades repetitivas. Como exemplo, é possível destacar projetos relativos às seguintes obras: rodovias, ferrovias, canais ou redes de tubulações, edifícios isolados (caso exista um pavimento-tipo), conjuntos habitacionais (cada casa ou bloco é uma unidade repetitiva) e pontes.

Moselhi e El-Rayes (1993) classificam os projetos repetitivos, ou lineares, em dois grupos: os projetos *típicos*, representados por projetos onde as atividades repetitivas têm a mesma duração em todos os estágios de construção, a exemplo de conjunto habitacionais, e *não-típicos*, cujas durações das atividades variam, como acontece em projetos de rodovias, onde o tempo de escavação pode variar de acordo com a região do terreno, variando devido à topografia do local. A produtividade pode se alterar também em razão do tipo de solo escavado, razões climáticas ou ser influenciada pelo aprendizado das equipes.

A maioria dos projetos de construção repetitivos pode ser classificada como não-típica. A variação da duração de uma atividade, de uma etapa da construção para outra, geralmente pode ser atribuída também a variações nas quantidades de trabalho e/ou produtividade das equipes. Assim, o planejamento eficiente e a programação de projetos repetitivos exigem continuidade do trabalho dos operários. Para que esta condição seja mantida, é preciso assegurar-se de que haverá movimentação ininterrupta das equipes de uma etapa para a próxima. O principal objetivo da manutenção da continuidade do trabalho é a maximização da utilização de recursos, fornecendo um cronograma que não apenas evita a ociosidade das equipes, mas também melhora sua curva de aprendizado (MOSELHI; EL-RAYES, 1993).

Tradicionalmente, as técnicas de rede, como o CPM, têm sido usadas no setor de construção para programar e controlar qualquer tipo de projeto. Essas técnicas, no entanto, apresentam grandes desvantagens quando aplicadas na programação de projetos repetitivos. Tais métodos requerem um grande número de atividades para representar um projeto repetitivo, dificultando, assim, a visualização da grande quantidade de dados envolvidos pelos profissionais encarregados (AGRAMA, 2012).

Para Reda (1990), o uso da técnica de rede CPM para programar projetos repetitivos tem três grandes desvantagens. Primeiro, como já citado, requer um grande número de atividades para representar o projeto. Esse grande número de atividades torna extremamente difícil visualizar a lógica da construção. Além disso, é preciso um computador de grande capacidade para processar os dados e fornecer a programação para cada atividade.

Em segundo lugar, a técnica CPM não garante a manutenção da continuidade do trabalho. Ou seja, uma equipe com ritmo de produção rápido pode ficar ociosa enquanto espera que as equipes anteriores, com taxas de produção mais lentas, concluam seu trabalho. Isso ocorre porque os planejamentos de CPM iniciam cada atividade só quando todas as suas predecessoras são concluídas. Por fim, para minimizar o custo do projeto, as técnicas de CPM encurtam apenas as atividades críticas, aumentando os recursos alocados a elas e, assim, aumentando suas taxas de produtividade. Isso pode resultar em taxas distintas para atividades semelhantes em diferentes etapas da construção (REDA, 1990).

De acordo com Reda (1990), nestes projetos, os recursos necessários para a execução das atividades passam de uma etapa para a outra. A palavra “etapa” é usada aqui para designar os estágios repetitivos, como os pavimentos-tipo em um prédio ou casas de um conjunto habitacional. A quantidade de recursos para cada atividade deve ser, então, cuidadosamente selecionada para atingir os seguintes objetivos:

- Manter uma taxa de produção constante para as equipes em cada atividade durante o projeto.
- Manter a continuidade do trabalho de cada equipe ao passar de uma etapa para a outra, eliminando assim o tempo ocioso esperando que as equipes anteriores terminem seu trabalho.
- Permitir intervalos de tempo (*buffers*) entre atividades na mesma etapa – por exemplo, um intervalo entre equipes de concretagem e de colocação de formas, de modo a esperar a cura do concreto.
- Permitir intervalos entre atividades em etapas diferentes – por exemplo, a execução da alvenaria deve ser iniciada dois pisos abaixo do andar onde ocorre a concretagem, para que o concreto possa alcançar a rigidez necessária.



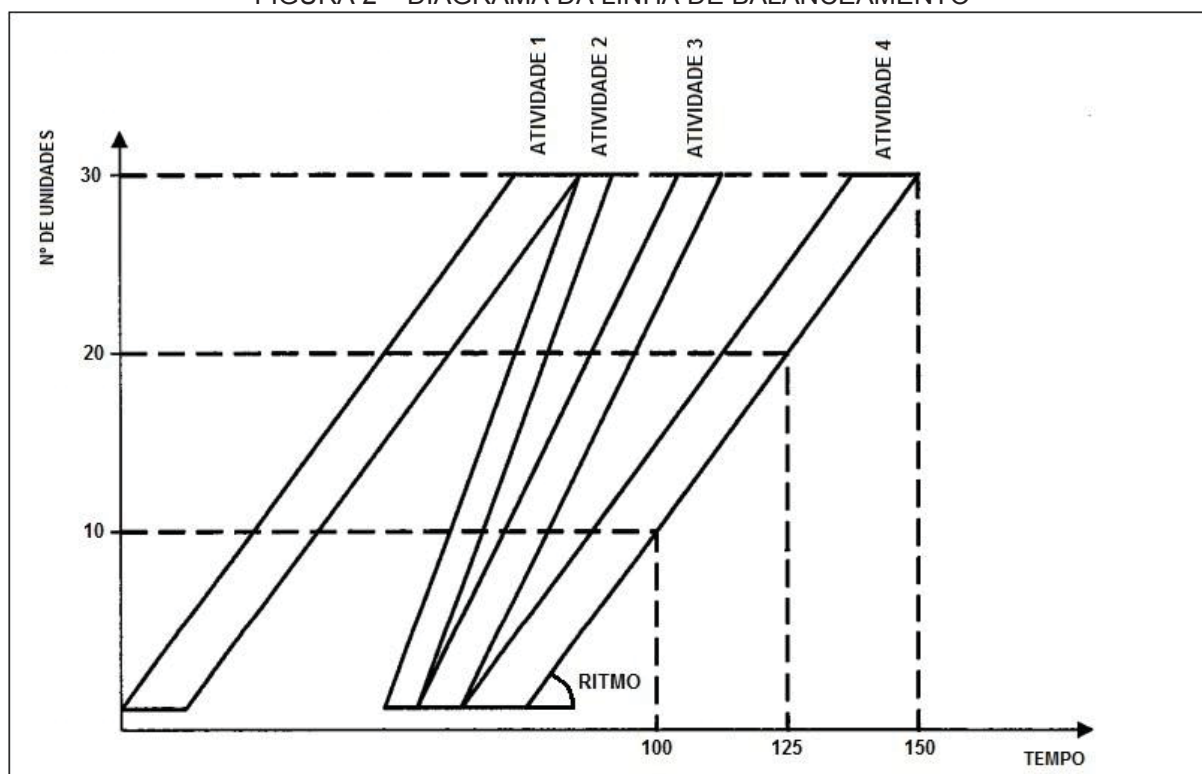
- Terminar o projeto com o mínimo custo possível, obedecendo a uma duração pré-determinada.

Com o reconhecimento das desvantagens dos métodos de rede, várias técnicas tradicionais de planejamento, além de uma variedade de técnicas especiais, foram desenvolvidas durante o século passado (AGRAMA, 2012). Entre elas, é possível citar a Linha de Balanceamento, o *Linear Scheduling Method* e o *Vertical Production Method*, que constituem abordagens gráficas, onde as atividades que se repetem são plotadas como linhas com inclinações constantes ou variáveis (onde a inclinação representa a taxa de produção), sendo representados nos eixos o progresso da obra e o tempo de execução (Figura 2). As vantagens dessas técnicas gráficas são a simplicidade e a facilidade de visualizar todo o projeto. Além disso, a continuidade do trabalho e a taxa de produção de cada equipe são mantidas (REDA, 1990).

Dentre as abordagens citadas, o método da Linha de Balanceamento é o mais utilizado no mercado de construção. Sua formulação é simples, buscando manter a continuidade do trabalho ao longo das unidades repetitivas, e serviu como ponto de partida para a maioria das formulações matemáticas desenvolvidas para o planejamento de projetos repetitivos desde a década de 1960 (AGRAMA, 2012). Nesta abordagem, a seleção de uma quantidade específica de recursos para uma atividade de construção determina a duração da atividade e o custo direto associado. A duração da atividade pode ser reduzida pela alocação de mais recursos com custo direto adicional (REDA, 1990).

A quantidade de trabalho necessária para concluir uma atividade é conhecida e fixa, sendo a taxa de produtividade, portanto, uma função tanto da duração da atividade como da quantidade de recursos alocados. O objetivo é finalizar o projeto dentro de uma duração prevista e com um custo direto mínimo, ao passo que se mantém as taxas de produtividade e a continuidade do trabalho de cada equipe constantes (REDA, 1990).

FIGURA 2 – DIAGRAMA DA LINHA DE BALANCEAMENTO



Fonte: Arditi, Tokdemir e Suh (2002)

Apesar do método da Linha de Balanceamento já ser bastante difundido no mercado, sua aplicação ainda requer muito tempo de trabalho. Os gerentes de projetos ainda precisam de um longo período para planejar efetivamente uma obra por meio desta técnica, o que atualmente constitui uma grande dificuldade, pois é muitas vezes necessário programar diversos projetos ao mesmo tempo, que tem que ser entregues em prazos cada vez menores. Assim, a pesquisa sobre métodos computacionais de otimização se torna relevante, uma vez que há grande demanda pela automatização dos processos de planejamento (MATHEW et al., 2016).

No Anexo I, a maioria das técnicas identificadas foram aplicadas ao processo de organização ou sequenciamento de atividades, usando como modelo matemático para a programação o Problema da Programação de Projetos com Restrições de Recursos. Assim, é necessário conceitua-lo mais detalhadamente, para em seguida apresentar outros conceitos importantes para a condução do método abordado na presente dissertação.

## 2.3 O PROBLEMA DA PROGRAMAÇÃO DE PROJETOS COM RESTRIÇÕES DE RECURSOS

O planejamento e a programação são comuns a muitos domínios de engenharia diferentes. Em um ambiente de fábrica, determinar quais tarefas devem ser executadas em quais máquinas, por quais os funcionários, podem fazer diferença entre um lucro significativo e o prejuízo total. No processo de desenvolvimento de *software*, atribuir responsabilidade pelas tarefas e gerenciar interrupções de maneira eficaz pode significar a diferença entre um produto que é lançado a tempo de suprir uma demanda de mercado e um produto que perde esse valor, pois algum outro já foi lançado antes (WALL, 1996).

De uma maneira geral, o processo de programação de projetos com restrições de recursos busca realizar a seguinte tarefa: dado um número de atividades, um conjunto de recursos e uma meta de desempenho, qual é a melhor maneira de atribuir os recursos às atividades de forma que o desempenho seja maximizado? O problema de programação geral (*Project Scheduling Problem*) engloba muitas variações, como os problemas de *job-shop* e de *flow-shop*, a programação de produção e o problema de programação de projetos com restrições de recursos (*Resource-Constrained Project Scheduling Problem - RCPSP*) (WALL, 1996).

Segundo Kolisch e Hartmann (1999) O RCPSP pode ser formulado da seguinte forma:

“Um único projeto consiste em um conjunto  $J = \{0, 1, \dots, n, n + 1\}$  de atividades que precisam ser processadas. Atividades fictícias 0 e  $n + 1$  correspondem o ‘início do projeto’ e ao ‘final do projeto’, respectivamente. As atividades são inter-relacionadas por dois tipos de restrições. Primeiro, restrições de precedência forçam uma dada atividade  $j$  a não ser iniciada antes que todas as suas atividades predecessoras imediatas contidas no conjunto  $P_j$  sejam finalizadas. Segundo, a execução das atividades requer recursos com capacidades limitadas. Tem-se  $K$  tipos de recursos, contidos no conjunto  $K = \{1, \dots, K\}$ . Enquanto estiver sendo processada, a atividade  $j$  requer  $r_{j,k}$  unidades do tipo de recurso  $k$  durante toda a sua duração  $d_j$ . O tipo de recurso  $k$  tem uma capacidade limitada de  $R_k$  durante todo o período do projeto. Assume-se que os parâmetros  $p_j$ ,  $r_{j,k}$  e  $R_k$  são determinísticos; para o início e o fim do projeto, temos  $d_j = 0$  e  $r_{j,k} = 0$  para todo  $k \in K$ . O objetivo do RCPSP é encontrar tempos de finalização possíveis, levando em consideração as precedências e restrições de recursos, de modo que a duração (*makespan*) do projeto seja minimizada”

Assim, se chamarmos de  $F_j$  o tempo de finalização da atividade  $j$ , a programação do projeto pode ser representada por um vetor de tempos de finalização

$(F_1, F_2, \dots, F_n)$ . Sendo  $A(t) = \{j \in J \mid F_j - p_j \leq t < F_j\}$  o conjunto de atividades sendo processadas no instante  $t$ , é possível então formular o problema matematicamente da seguinte forma (KOLISCH; HARTMANN, 1999):

$$\text{Minimize } F_{n+1} \quad (1)$$

Sujeito a:

$$F_i \leq F_j - d_j \quad j = 0, 1, \dots, n + 1; i \in P_j \quad (2)$$

$$\sum_{j \in A(t)} r_{j,k} \leq R_k \quad k \in K; t \geq 0 \quad (3)$$

$$F_j \geq 0 \quad j = 1, \dots, n + 1 \quad (4)$$

A equação 1 representa então a função-objetivo do problema, que consiste em minimizar o tempo de finalização da atividade final do projeto, diminuindo, portanto, sua duração ou *makespan*. A restrição descrita na equação 2 diz respeito às relações de precedência entre duas atividades. A equação 3, por sua vez, descreve a limitação da quantidade de recursos utilizados em cada instante de tempo. Por fim, a equação 4 define o domínio das variáveis de decisão (KOLISCH; HARTMANN, 1999)

Em geral, os problemas de programação de projetos são NP-difíceis, o que significa que não há algoritmos conhecidos para encontrar soluções ótimas em tempo polinomial. Algoritmos existem para resolver de maneira exata algumas formas do problema, mas eles geralmente demandam um longo tempo de processamento à medida que o tamanho do problema aumenta ou quando novas restrições são adicionadas. Como resultado, a maioria das pesquisas tem se dedicado a simplificar o problema de programação até o ponto em que alguns algoritmos podem encontrar soluções ou a elaborar heurísticas eficientes para encontrar boas soluções. Em alguns casos, o problema pode limitar-se simplesmente em encontrar uma solução viável e, muitas vezes, a existência de uma solução ótima específica não pode ser assegurada (WALL, 1996).

Assim, para resolver problemas de programação reais, pesquisadores desenvolvem procedimentos heurísticos baseados principalmente em algoritmos evolucionários (GHODDOUSI et al., 2013; BETTEMIR; SONMEZ, 2014; LIU; AL-HUSSEIN; LU, 2015; GIRAN; TEMUR; BEKDAS, 2016; EL-ABBASY; BIRUK; JASKOWSKI, 2017; ELAZOUNI; ZAYED, 2017), programação linear inteira (MENESI;

HEGAZY, 2014) e otimização de redes (POTGIETER; VAN ROOYEN, 2016). Segundo Ichihara (1999), a maioria das heurísticas emprega regras de prioridade para programar as atividades, as quais permitem solucionar conflitos de recursos por meio da programação de atividades com maior prioridade, no menor tempo de disponibilidade do recurso restrito.

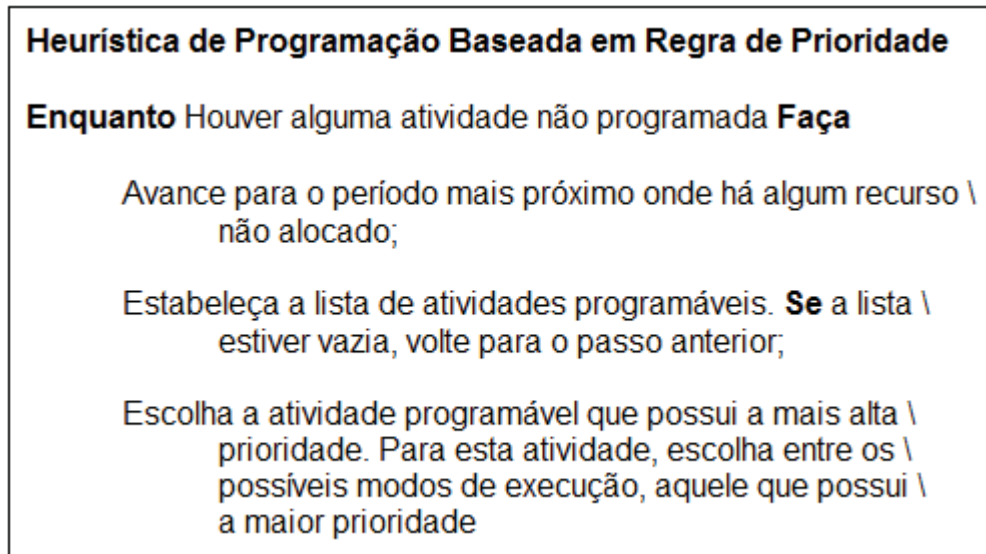
O RCPSP, por sua vez, pertence à classe dos problemas combinatoriais, a qual é caracterizada pelo crescimento fatorial do tempo computacional requerido para solucionar todas as possíveis soluções, de acordo com o aumento do tamanho do problema. Assim, as técnicas para solucioná-los pertencem à área de Otimização Combinatorial. Os procedimentos heurísticos como programação baseada em regras de prioridade e os algoritmos evolucionários (ICHIHARA, 1999), como também métodos computacionais inspirados em fenômenos da físico-química (GOLDBARG; GOLDBARG; LUNA, 2016) são ferramentas de otimização combinatorial largamente utilizadas na busca por soluções para o RCPSP, motivo pelo qual torna-se oportuno conceitua-las mais profundamente.

#### 2.4.1 Heurísticas de Programação Baseadas em Regras de Prioridade

Segundo Kolisch (1996), as técnicas de solução heurísticas baseadas em regras de prioridade ainda estão entre os mais importantes métodos de programação para resolver o RCPSP, em razão do seu caráter intuitivo e de fácil utilização, tornando satisfatório seu uso em pacotes comerciais, como também por exigir pouco esforço computacional, podendo integrar aos métodos de busca local da área de Inteligência Artificial.

No funcionamento destes processos heurísticos, uma dada atividade é considerada programável se duas condições forem satisfeitas: todas as suas atividades predecessoras já estão programadas e se ela possui pelo menos um modo de execução possível. Um modo de execução só é considerado possível se cada recurso requerido é menor ou igual à soma do recurso disponível para o período de tempo considerado (ICHIHARA, 1999). O procedimento computacional é descrito na Figura 3.

FIGURA 3 – ESTRUTURA EM PSEUDOCÓDIGO DE HEURÍSTICA DE PROGRAMAÇÃO BASEADA EM REGRA DE PRIORIDADE



Fonte: adaptado de Ichihara (1999)

Um procedimento heurístico de programação baseado em regras de prioridade é, geralmente, composto por dois elementos: um Esquema de Geração de Programação e uma regra de prioridade. Existem dois tipos de esquema possíveis: o método Serial e o Paralelo. Ambos geram uma programação para o projeto por meio de um processo onde uma programação parcial, onde tempos de finalização foram atribuídos apenas a uma parte do total de atividades, vai sendo estendida à medida que os tempos finais de cada tarefa vão sendo incluídos. Em cada estágio, o esquema considera um conjunto de atividades programáveis, chamado de *conjunto de decisão*. Uma regra de prioridade é, então, empregada a fim de escolher uma atividade contida no conjunto, que é inserida na programação (KOLISCH, 1996).

Os métodos Serial e Paralelo, descritos a seguir, correspondem a um procedimento de passo único, ou seja, onde apenas um esquema de geração de programação e uma regra de prioridade é empregada, gerando apenas uma alternativa de programação (KOLISCH; HARTMANN, 1999). Em processos de múltiplos passos, realizam-se vários únicos passos para gerar uma amostra de possíveis soluções diferentes, onde a melhor é escolhida (ICHIHARA, 1999).

#### 2.4.1.1 O Método Serial

O esquema de geração de programação serial consiste em  $n = 1, \dots, J$  estágios, onde, em cada um deles, uma atividade é selecionada e programada para a

menor data de término possível, levando em consideração suas precedências e requisitos de recursos, como descrito na Figura 3 (KOLISCH; HARTMANN, 1999).

FIGURA 4 – PSEUDOCÓDIGO DO ESQUEMA DE PROGRAMAÇÃO SERIAL

**Esquema de Programação Serial**

**Inicialização**  
 $F_0 = 0$   
 $S_0 = \{0\}$   
 $n = 1; \{n = 1, \dots, J\}$

**Enquanto**  $|S_n| < |g|$  **faça** Estágio  $n$   
    Calcular  $D_n, F_n, R_k(t)$  ( $k \in K; t \in F_n$ )  
    Selecionar uma atividade  $j \in D_n$   
     $FC_j = \max \{F_i / i \in P_j\} + d_j$   
     $F_j = \min \{t / FC_j \leq t \leq FT_j\}$ , com  $\{r_{j,k} \leq R_k(t)\} \setminus$   
     $\{l_j = t - d_j + 1, \dots, t\}, \{r \in R\}$   
     $S_{n+1} = S_n \cup \{j\}$   
     $n = n + 1$

Onde:

$d_j$ : duração da atividade  $j$ .  
 $D_n$ : conjunto de decisão;  $D_n = \{j / j \notin S_n, P_j \subseteq S_n\}$   
 $F_n$ : data de término da atividade  $j$ .  
 $FC_j$ : data mais cedo possível para o término da atividade  $j$ .  
 $FT_j$ : data mais tarde possível para o término da atividade  $j$ .  
 $S_n$ : conjunto de atividades programadas.  
 $P_j$ : conjunto de predecessoras da atividade  $j$ .  
 $l_j$ : data programada de início da atividade  $j$ .  
 $R_k(t)$ : quantidade do recurso  $k$  disponível no tempo  $t$ .  
 $R_k(t) = \sum_{j \in A(t)} R_k - r_{k,j}$   
 $A(t)$ : conjunto das atividades programadas no tempo  $t$ .  
 $R_k$ : nível de disponibilidade de recurso  $k$ .  
 $r_{k,j}$ : quantidade do recurso  $r$  utilizado pela atividade  $j$ .

Fonte: adaptado de Kolisch (1996).

Associado com cada estágio estão dois conjuntos de atividades: o conjunto  $S_n$ , onde encontram-se as atividades já programadas, e o conjunto de decisão  $D_n$ , que contém as atividades não programadas, cujas predecessoras já estão no conjunto  $S_n$ . Em cada estágio, uma atividade  $j$  do conjunto de decisão é selecionada por meio de uma regra de prioridade e programada para iniciar na data mais cedo  $l_j$  que satisfaça as restrições de precedências e de recursos; no caso de duas atividades possuírem mesma prioridade, será prioritária a atividade de menor número. Depois, a atividade selecionada é removida de  $D_n$  e colocada em  $S_n$  e as atividades que já possuem todas as precedentes já programadas, podem entrar em  $D_n$ . O algoritmo termina no estágio



$n=J$ , quando todas as atividades estiverem programadas (KOLISCH, 1996; KOLISCH; HARTMANN, 1999).

Em uma rede do tipo atividade-no-nó,  $FC_j$  é a data mais cedo que uma atividade pode terminar, respeitando-se o cumprimento de todas as precedências;  $FT_j$  é a data mais tarde que uma atividade pode terminar, de forma a não exceder o prazo  $T$  fixado para o término do projeto;  $F_j$  representa a data programada para o final da atividade, consistindo na menor data entre  $FC_j$  e  $FT_j$ , em que a disponibilidade de recursos  $R_k$  não tenha sido ultrapassada (ICHIHARA, 1999).

#### 2.4.1.2 O Método Paralelo

O método paralelo também consiste em  $J$  estágios. Em cada uma dessas etapas, tem-se um conjunto de atividades a ser programado. A característica que diferencia o método paralelo, é que cada estágio  $n$  é associado com um tempo de programa  $t_n$  (*schedule time*), onde  $t_m \leq t_n$  para  $m < n$ . Por “tempo de programa” entende-se o tempo para o qual a finalização da atividade está definida. Por causa deste tempo de programa, o conjunto de atividades programadas é dividido em dois subconjuntos: o conjunto completo  $C_n$ , que contém as atividades que estavam programadas e encontram-se finalizadas até o tempo de programa; e o conjunto ativo  $A_n$ , que contém as atividades que estavam programadas, mas que ainda estão ativas no tempo  $t_n$ .

O conjunto de decisão  $D_n$ , diferentemente do que acontece no Método Serial, contém todas as atividades que ainda podem ser programadas, levando em consideração não só as restrições de precedência, como também as de recursos. O programa parcial de cada estágio é feito pelas atividades que estão no conjunto completo e no conjunto ativo. Cada estágio é feito em dois passos: (1) o tempo de programa novo é determinado e as atividades com um tempo final igual ao novo tempo de programa são removidas do conjunto ativo e colocadas no conjunto completo; (2) uma atividade do conjunto de decisão é selecionada com a regra de prioridade e programada para iniciar no tempo de programa corrente. Depois, esta atividade é removida do conjunto de decisão e colocada no conjunto ativo. O passo (2) é repetido até que o conjunto de decisão esteja vazio. O Método Paralelo termina quando todas as atividades estão no conjunto ativo (Figura 5).

FIGURA 5 – PSEUDOCÓDIGO DO ESQUEMA DE PROGRAMAÇÃO PARALELO

**Esquema de Programação Paralelo**

**Inicialização**  
 $n = 1; \{ n = 1, \dots, J \}$   
 $t_n = 0$   
 $D_n = \{ 1 \}$   
 $A_n = C_n = \{ 0 \}$   
 $R_k(t) = R_k \quad \forall r \in R$

**Vá Para o Passo 2**

**Enquanto**  $|A_n \cup C_n| < J$  **faça** Estágio  $n$

(1)  $t_n = \min \{ F_j \mid j \in A_{n-1} \}$   
 $A_n = A_{n-1} \mid \{ j \mid j \in A_{n-1}, F_j = t_n \}$   
 $C_n = C_{n-1} \cup \{ j \mid j \in A_{n-1}, F_j = t_n \}$   
 Calcular  $R_k(t) \quad \forall r \in R \text{ e } D_n$

(2) Selecionar uma atividade  $j \in D_n$   
 $F_j = t_n + d_j$   
 $A_n = A_n \cup \{ j \}$   
 Calcular  $R_k(t) \quad \forall r \in R \text{ e } D_n$   
**Se**  $D_n \neq \{ \}$  **Então Vá Para o Passo 2** **Senão**  $n = n+1$

Onde:

$A_n$  : conjunto de atividades ativas no tempo  $t_n$   
 $C_n$  : conjunto das atividades completadas até o tempo  $t_n$   
 $d_j$  : duração da atividade  $j$   
 $D_n$  : conjunto de decisão  
 $D_n = \{ j \mid j \notin \{ A_n \cup C_n \}, P_j \subseteq C_n, R_k(t) \leq R_k \quad \forall r \in R \}$   
 $F_j$  : data programada para o fim da atividade  $j$   
 $P_j$  : conjunto de predecessoras da atividade  $j$   
 $R_k(t)$  : quantidade de recurso  $k$  disponível no tempo  $t$   
 $R_k(t) = \sum_{j \in A(t)} R_k - r_{k,j}$   
 $R_k$  : nível de disponibilidade do recurso  $k$   
 $r_{k,j}$  : quantidade de recurso  $r$  utilizado pela atividade  $j$

Fonte: adaptado de Kolisch (1996)

#### 2.4.2. Computação Inspirada na Físico-química

Métodos computacionais que buscam copiar características de fenômenos da natureza fazem parte do campo da informática denominado Computação Natural, que engloba diversas áreas como Simulação Computacional de Sistemas Naturais, Computação Bioinspirada, Computação com Materiais Naturais e a Computação Inspirada na Físico-química (GOLDBARG; GOLDBARG; LUNA, 2016).

Existem também diversas técnicas dentro da área de Computação Inspirada na Físico-química, sendo uma das mais conhecidas e utilizadas o *Simulated Annealing* (SA), que consiste em uma forma de copiar o processo termodinâmico de

recozimento de materiais como metais e vidros (GOLDBARG; GOLDBARG; LUNA, 2016).

O *Simulated Annealing* é uma meta-heurística de caráter probabilístico, inspirada em processo controlado de resfriamento lento de metais e parte da noção de que, no início de um processo de resfriamento, quando a temperatura da substância é alta, suas moléculas estão desorganizadas, formando um arranjo caótico. No entanto, à medida que o metal é lentamente resfriado e sua temperatura diminui, o movimento das moléculas começa a ficar limitado, fazendo com que elas tendam a se organizar de maneira simétrica, formando cristais (BETTEMIR, 2009; LUKE, 2012).

Desse modo, se o resfriamento for feito de forma homogênea por toda a peça de metal, as moléculas têm a oportunidade de se redistribuir também homogeneamente, reduzindo eventuais defeitos. Quanto mais lento for o esquema de resfriamento, mais provavelmente as configurações serão cristalizadas em patamares de baixa energia implícita. Caso contrário, serão formadas configurações típicas de altas temperaturas, com moléculas dispostas caoticamente, mesmo à temperatura ambiente (GOLDBARG; GOLDBARG; LUNA, 2016).

A ideia geral desta meta-heurística é solucionar um problema de minimização em analogia com um esquema de aquecimento/resfriamento que permita a formação de configurações atômicas de baixa energia em um metal. Assim, no SA, o movimento realizado pelo algoritmo de busca ao passar de uma solução para a outra é um processo probabilístico controlado por equações do modelo de Metropolis et al. (1953), sendo a temperatura o principal parâmetro de controle da busca (GOLDBARG; GOLDBARG; LUNA, 2016).

De modo geral, o algoritmo inicia com uma solução aleatória, que a cada iteração é alterada. O diferencial desta técnica é que ela utiliza um mecanismo de decisão probabilística para definir se a solução inicial será substituída ou não pela nova solução gerada pela alteração. Em analogia ao processo termodinâmico, a variação da “energia”, ou seja, da qualidade das soluções é avaliada. Caso a nova solução tenha melhor qualidade, esta substitui a anterior. Caso contrário, a nova solução ainda pode ser aceita sob uma certa probabilidade, que consiste numa simplificação da distribuição de probabilidade proposta por Maxwell-Boltzmann e que

está associada a transformações termodinâmicas (LUKE, 2012; GOLDBARG; GOLDBARG; LUNA, 2016):

$$P(t, Q_i, Q_{i+1}) = e^{\frac{Q_i - Q_{i+1}}{t}} \quad (5)$$

Onde:

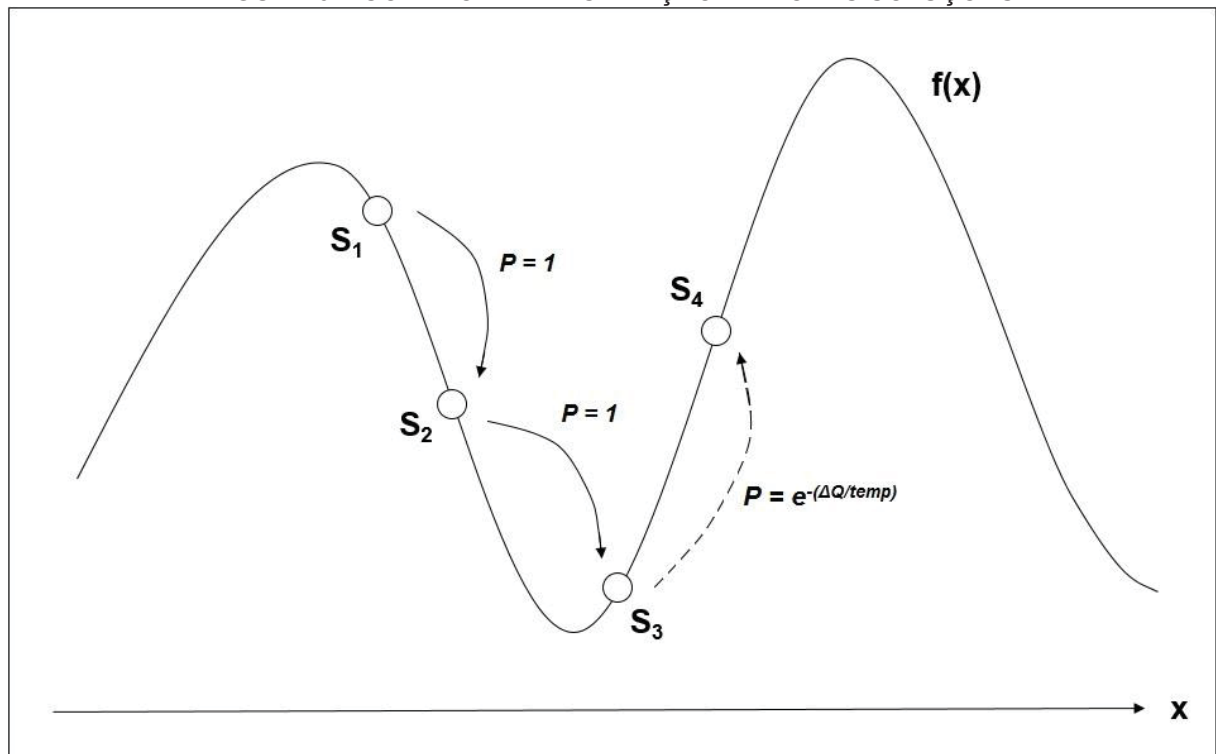
$Q_i$  = qualidade ou aptidão da solução inicial.

$Q_{i+1}$  = qualidade da solução alterada.

$t$  = temperatura do sistema, sempre positiva.

Portanto, é possível ainda que o algoritmo aceite soluções piores durante o processo de busca. A equação 5 se mostra interessante em dois sentidos: primeiramente, deve-se notar que o expoente é necessariamente um valor negativo, pois  $Q_i$  é pior que  $Q_{i+1}$ . Assim, se  $Q_i$  for muito pior que  $Q_{i+1}$ , a fração assume um valor elevado, resultado em uma probabilidade próxima de 0. Caso  $Q_i$  seja um valor muito próximo de  $Q_{i+1}$ , a probabilidade fica próxima de 1. Desse modo, quando  $Q_i$  não for um valor *muito* pior que  $Q_{i+1}$ , ainda será possível aceitá-lo com certa probabilidade. Por fim ainda existe o parâmetro  $t$ , análogo a temperatura de resfriamento. Quando este valor é próximo de 0, a fração assume um valor alto, resultado em uma probabilidade também próxima de zero. Se  $t$  for um valor alto, a probabilidade, por sua vez, fica próxima de 1. A ideia é definir a temperatura inicial em um valor alto, o que faz com que o algoritmo aceite qualquer solução criada, independentemente de sua qualidade, configurando um processo de busca aleatória (Figura 6). Em seguida a temperatura é reduzida gradualmente, de forma que as soluções ruins sejam cada vez menos aceitas (LUKE, 2012).

FIGURA 6 – CONTROLE DE ACEITAÇÃO DE NOVAS SOLUÇÕES



Fonte: adaptado de Goldbarg, Goldbarg e Luna (2016)

A taxa de redução da temperatura é chamada de esquema de resfriamento. Quanto mais lenta for a diminuição da temperatura, mais elevada será a capacidade de exploração do espaço de busca pelo algoritmo (LUKE, 2012). Os passos executados pelo algoritmo podem ser descritos em pseudocódigo de acordo com a Figura 7.

FIGURA 7 – ESTRUTURA BÁSICA EM PSEUDOCÓDIGO DO *SIMULATED ANNEALING*

**Simulated Annealing**

- (1) Definir temperatura inicial  $t = T_0$
- (2) Definir solução aleatória inicial  $S_i$
- (3) **Enquanto**  $T >$  número próximo de zero:
  - Gerar nova solução  $S_{i+1}$
  - Calcular  $\Delta Q = \text{Qualidade}(S_{i+1}) - \text{Qualidade}(S_i)$
  - Se**  $\Delta Q < 0$ :
    - $S_i = S_{i+1}$
  - Caso contrário se**  $\text{RANDOM}(0,1) > e^{-(\Delta Q / t)}$ :
    - $S_i = S_{i+1}$
- (4) Diminuir a temperatura

Fonte: adaptado de Goldbarg, Goldbarg e Luna (2016).

### 2.4.3. Computação Evolucionária

Segundo Ichihara (1999), a partir da década de 1950, pesquisadores da área de ciência da computação estudaram os sistemas evolucionários naturais com a ideia de que a evolução poderia ser utilizada como uma ferramenta para resolver os problemas de engenharia. Surgiu, então, a Computação Evolucionária, que se fundamenta no princípio de que os indivíduos melhor adaptados, possuem maior capacidade de sobrevivência no mesmo ambiente.

De acordo com Goldbarg, Goldbarg e Luna (2016), a computação evolucionária é caracterizada pelas seguintes condições:

- É realizada através de um processo iterativo.
- Baseia-se em uma população.
- Possui intrinsecamente uma arquitetura de processamento paralelizável.
- Corresponde a um processo de busca estocástica com viés – busca guiada.
- Emprega o princípio darwiniano da seleção natural de acúmulos de variações genéticas.

Os algoritmos evolucionários pesquisam a melhor solução por meio da melhoria contínua da *população*, formada por representações de soluções potenciais  $X^i$  ( $i=1, \dots, p$ ), denominadas *indivíduos* ou *cromossomos* (Figura 8). Cada um desses elementos caracteriza-se como um ponto no espaço de busca, ao qual é atribuída uma medida de qualidade calculada por uma função-objetivo (ICHIHARA, 1999)

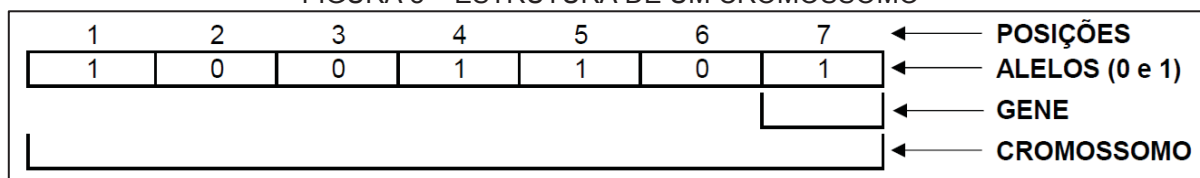
Cada evolução de uma população é denominada *geração*, que é composta por ciclos de: *avaliação*, realizada pela função-objetivo; *seleção*, baseada na determinação da *aptidão* (*fitness*) dos indivíduos; e *reprodução*, que consiste na criação de descendentes por meio de *cruzamentos* e *mutações* de indivíduos selecionados (ICHIHARA, 1999). Os termos mais usados na área são reunidos na Tabela 1.

Tabela 1 – Termos mais utilizados em Computação Evolucionária

|                                   |  |
|-----------------------------------|--|
| <b>Indivíduo</b>                  | Uma solução possível para o problema (solução candidata)   |
| <b>Pai e filho</b>                | Um filho é um cópia alterada de uma solução possível (pai)                                       |
| <b>População</b>                  | Conjunto de soluções possíveis   |
| <b>Aptidão ou <i>fitness</i></b>  | Qualidade de um solução (medida pela função-objetivo do problema)                                |
| <b>Seleção</b>                    | Escolha de indivíduos baseada em sua aptidão   |
| <b>Mutação</b>                    | Alteração das características de uma solução aleatoriamente                                      |
| <b>Cruzamento ou Recombinação</b> | Criação de duas novas soluções a partir da combinação de características de duas soluções “pais” |
| <b>Genótipo</b>                   | Estrutura de dados de um indivíduo   |
| <b>Cromossomo</b>                 | A representação do genótipo na forma de um vetor de módulo fixo                                  |
| <b>Gene</b>                       | Divisão conceitual ou bloco funcional de um cromossomo (codifica características)                |
| <b>Alelo</b>                      | Característica ou valor numérico que representa um gene  |
| <b>Locus</b>                      | Posição em que o gene se encontra no cromossomo  |
| <b>Fenótipo</b>                   | A forma que o indivíduo opera durante a avaliação da aptidão                                     |
| <b>Geração</b>                    | População produzida em cada ciclo  |

Fonte: Ichihara (1999) e Luke (2015)

FIGURA 8 – ESTRUTURA DE UM CROMOSSOMO



Fonte: adaptado de Ichihara (1999)

Os indivíduos da população, são, portanto, propostas de solução para o problema em pauta (viáveis ou não). Por meio da avaliação dos indivíduos da população, o processo de reprodução é direcionado a favorecer os indivíduos de melhor adequação (os mais aptos), seguindo a lógica da evolução darwiniana. Na natureza, recombinações e mutações modificam o material genético dos indivíduos. No contexto computacional, caracterizam-se como procedimentos heurísticos que visam utilizar as possibilidades combinatórias codificadas nos indivíduos da população trabalhada pelo algoritmo, como esquematizado na Figura 9 (GOLDBARG; GOLDBARG; LUNA, 2016).



FIGURA 9 – ESTRUTURA BÁSICA DOS ALGORITMOS GENÉTICOS

### **Estrutura dos Algoritmos Evolucionários**

- (1) Define a codificação adequada para as soluções.
- (2) Gere uma população inicial  $P(0) = X^1, X^2, X^3, \dots, X^p$ ; faça iteração  $t = 0$
- (3) Avaliação: calcule a aptidão  $f(X^i)$  para cada indivíduo  $X^i$ ,  $i = 1, \dots, p$ .
- (4) Seleção: selecione os indivíduos de  $P(t)$ ; isto gera o grupo de acasalamento  $S(t)$ . O mesmo indivíduo de  $P(t)$  pode aparecer várias vezes em  $S(t)$ .
- (5) Reprodução: formar pares de indivíduos em  $S(t)$  e para cada par de indivíduos:
  - Recombine o par, com probabilidade  $c$ , e copie o descendente resultante em  $S(t+1)$ ; o par de cromossomos é eliminado e substituído pelo seu descendente.
  - Copie os pares de indivíduos no conjunto  $S(t+1)$  com probabilidade  $1 - c$ .Para cada indivíduo de  $S(t+1)$ :
  - Aplique a mutação no indivíduo com probabilidade  $m$  e copie os indivíduos transformados em  $P(t+1)$ ;
  - Copie o indivíduo em  $P(t+1)$  com probabilidade  $1 - m$ .
- (6) Faça  $t = t + 1$  e retorne para o Passo 2 até que o critério de parada seja alcançado.

Fonte: Pierreval e Tautou (1996)

Atualmente, as técnicas metaheurísticas diretamente associadas à computação evolucionária são principalmente: Algoritmos Meméticos, Algoritmos Transgenéticos, Algoritmos Culturais, Algoritmos Simbióticos e, principalmente, os Algoritmos Genéticos, foco deste trabalho e abordado no item a seguir.

## 2.5 ALGORITMOS GENÉTICOS

Os Algoritmos Genéticos (*Genetic Algorithms* - GAs) referem-se a uma classe de procedimentos de busca adaptativa, inspirada nos princípios evolutivos genéticos das populações na natureza. Sua denominação originou-se da analogia entre a representação de uma estrutura complexa através de um vetor de componentes e ideia de estrutura genética de um cromossomo. O algoritmo imita o curso da evolução natural, onde as populações se modificam através das gerações conforme os princípios de seleção natural e da sobrevivência dos mais aptos. Isso significa que os genes dos indivíduos altamente aptos influenciarão um número crescente de indivíduos a cada geração, e sua espécie evolui para se tornar cada vez mais adaptada ao seu ambiente (ICHIHARA, 1999).



Um AG simula esse processo, tomando uma população inicial de indivíduos e aplicando operadores genéticos artificiais em cada geração. Computacionalmente, cada indivíduo é codificado em um *string* ou cromossomo, que representa uma solução possível para determinado problema. A adaptação dos indivíduos é avaliada segundo uma função de aptidão (*fitness*). Basicamente, aos indivíduos altamente aptos (melhores soluções), são dadas maiores chances de se reproduzirem, trocando partes de informação genética, em um procedimento denominado Cruzamento ou Recombinação. Há também um operador de Mutação, que é utilizado para alterar alguns genes dos cromossomos, conferindo diversidade à população. A descendência, por sua vez, pode substituir toda a população atual ou apenas os indivíduos com menos aptidão. O ciclo de avaliação, seleção e reprodução é repetido até que uma solução satisfatória seja encontrada (ICHIHARA, 1999).

Os GAs foram pela primeira vez introduzidos por Holland (1962), e hoje constituem a metaheurística de maior sucesso para a solução de problemas de Otimização Combinatória. Os GAs clássicos são eficientes na solução de problemas combinatórios especialmente por sua enorme capacidade de diversificação (GOLDBARG; GOLDBARG; LUNA, 2016). Essa diversificação se reflete em cinco vantagens principais oferecidas pela técnica (ICHIHARA, 1999):

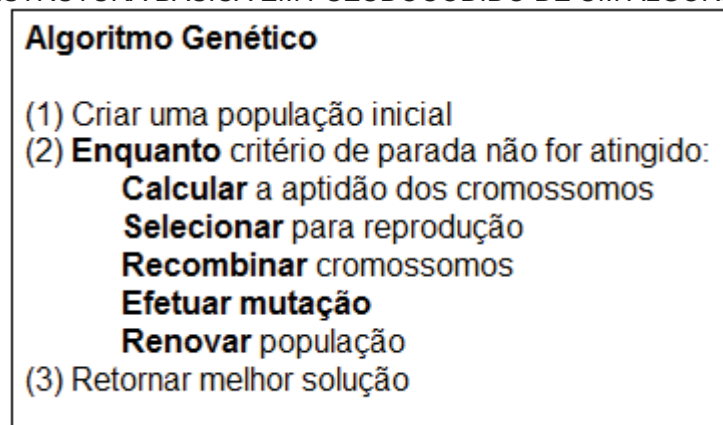
- **Independência de domínio:** os GAs trabalham sobre a codificação do problema, tornando possível a criação de programas computacionais gerais, capazes de resolver muitos tipos de problemas diferentes.
- **Não-linearidade:** as técnicas de otimização convencionais se baseiam, geralmente, em suposições irreais de linearidade, convexidade e diferenciabilidade, entre outras, enquanto GAs não as utilizam.
- **Robustez:** a independência de domínio e a não-linearidade tornam possível que os GAs resolvam grande variedade de problemas, bem como trabalhar com funções não-lineares.
- **Facilidade de modificação:** as modificações em um GA, a fim de adaptá-lo a tipos específicos de problemas são facilmente implementadas, diferente do que ocorre com a maioria das heurísticas.
- **Natureza paralela:** o GA facilita a efetivação de implementações com processamento paralelo, permitindo reduções no tempo de processamento.

Hillier e Lieberman (2010) definem a estrutura geral de algoritmo genético em três grandes etapas:

- **Inicialização:** crie uma população inicial de soluções experimentais viáveis, gerando-as aleatoriamente. Calcule o valor da função-objetivo, ou seja, a aptidão para cada membro da população atual.
- **Iteração:** use um processo aleatório que tende na direção dos membros mais adaptados da população atual para selecionar alguns deles (um número par) para se tornar “pais”. Emparelhe aleatoriamente os pais e depois faça com que cada par gere dois “filhos” (novas soluções experimentais viáveis), cujas características (genes) são uma mistura aleatória das características dos pais. Provoque mutações nos filhos sob determinada probabilidade. Retenha os filhos e um número suficiente dos melhores membros da atual população para formar uma nova população do mesmo tamanho na próxima iteração. Descarte os demais. Avalie a adaptação de cada novo membro (os filhos) da nova população).
- **Regra de parada:** use uma regra de parada, tal como um número fixo de iterações, uma quantidade fixa de tempo de processamento ou um número fixo de iterações consecutivas sem aprimoramento da melhor solução encontrada até então. Use a melhor solução experimental encontrada entre as iterações como solução final.

Em forma de pseudocódigo, a formulação geral dos GAs pode ser representada de acordo com a Figura 10.

FIGURA 10 – ESTRUTURA BÁSICA EM PSEUDOCÓDIGO DE UM ALGORITMO GENÉTICO



Fonte: adaptado de Goldberg, Goldberg e Luna (2016)

Desse modo, antes do algoritmo ser implementado, é necessário que se definam diversos parâmetros, correspondendo a perguntas como: Qual deve ser o tamanho da população? Como devem ser selecionados os “pais”? Quais características dos pais devem ser passadas para os “filhos”? Como as mutações devem mudar as características dos filhos? Qual regra de parada deve ser usada? (HILLIER; LIEBERMAN, 2010). Para melhor realizar essas definições, é preciso então conhecer os tipos de operadores genéticos e as possibilidades de operação para cada um deles, que serão definidos neste trabalho de acordo com Goldbarg, Goldbarg e Luna (2016).

### 2.5.1 Operadores de Cruzamento ou Recombinação

Como evidenciado anteriormente, o processo evolutivo descrito pelo algoritmo genético se inicia pela formação de uma população inicial. O tamanho da população é um parâmetro que deve ser fixado, em princípio, de forma experimental. Este constituinte do processo evolucionário regula a cobertura do espaço de busca. Assim, quanto maior for a população, mais informações sobre o espaço de busca estarão disponíveis, mas seu aumento também traz desvantagens quanto ao esforço computacional.

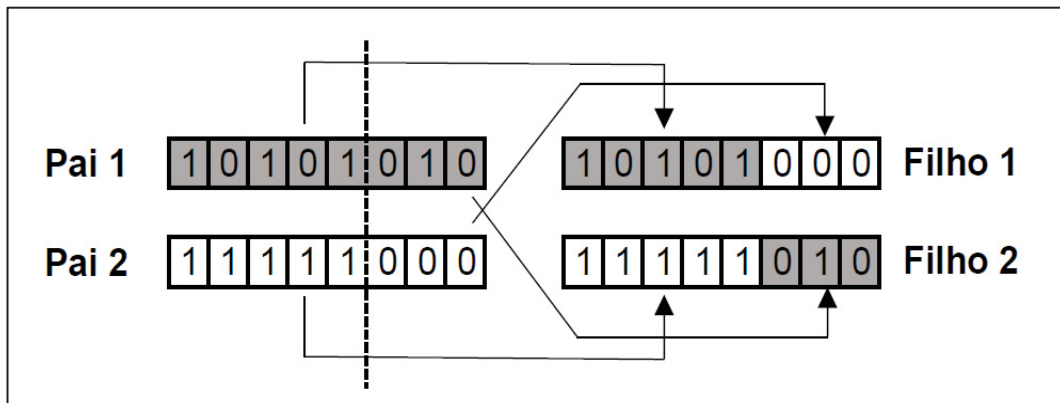
Com uma população formada e um critério de aptidão definido, ou seja, uma função-objetivo, é possível avaliar todos os indivíduos e aplicar um mecanismo de seleção para escolher um subconjunto para a reprodução. A partir desta etapa, são implementados os operadores genéticos, iniciando-se pelo Cruzamento ou Recombinação (*crossover*). Nesta ocasião, o código genético dos cromossomos deve ser misturado de modo que se formem novos cromossomos, denominados cromossomos filhos.

O processo de reprodução requer operadores específicos e uma conveniente formalização de como os trechos de DNA serão coletados dos pais e misturados para formarem novos indivíduos. Essas definições envolvem decisões como: quantos filhos serão formados, como os pais serão reunidos dois a dois, entre outras. Existem três operadores de cruzamento principais:

- **De um ponto:** o operador de um ponto (de quebra) divide o cromossomo em dois trechos. A quebra é caracterizada na Figura 11 pela linha tracejada

que divide os pais. Após a divisão, são formados quatro trechos de “DNA” que são, em seguida, combinados.

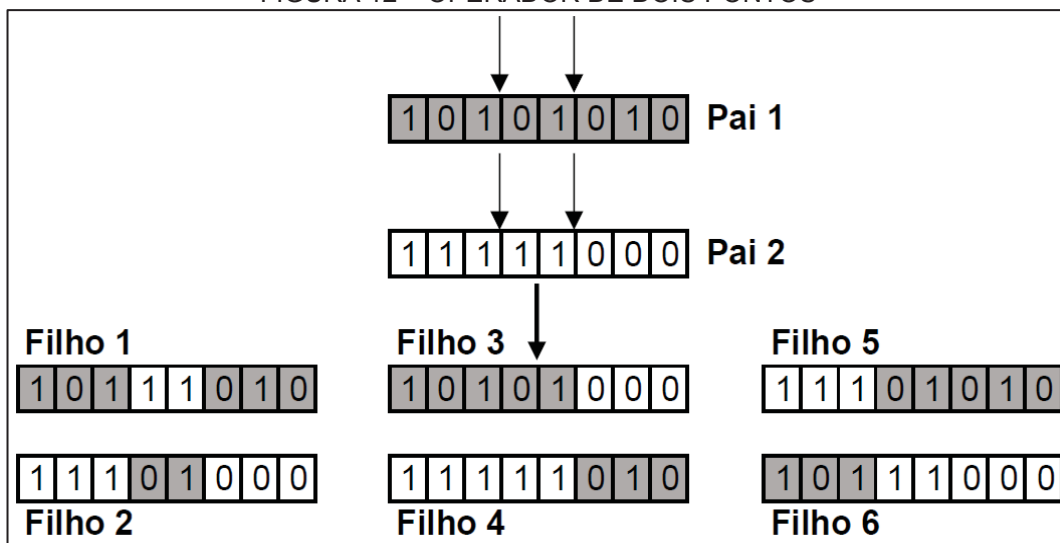
FIGURA 11 – OPERADOR DE UM PONTO



Fonte: Goldberg, Goldberg e Luna (2016)

- **De dois pontos:** como evidenciado na Figura 12, operador de dois pontos “quebra” o cromossomo em três regiões, sinalizadas pelas setas. Nesse caso, formam-se seis segmentos genéticos, sendo possível gerar vários filhos, dos quais são destacados seis possíveis.

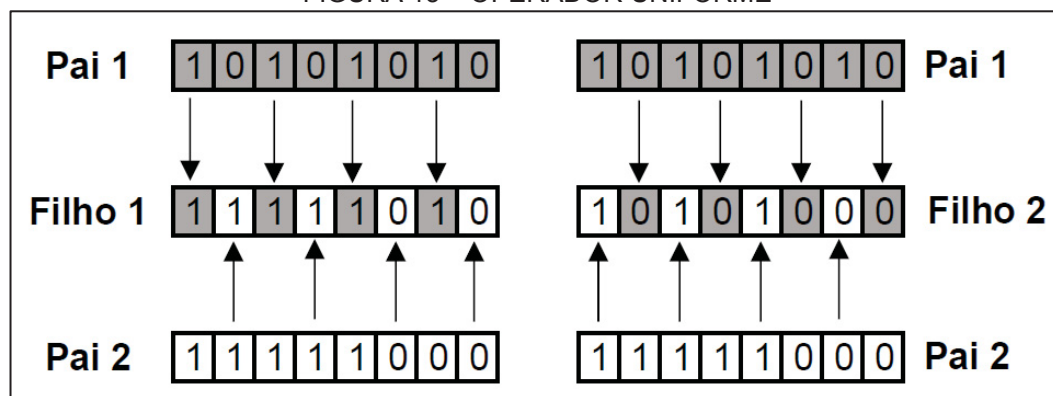
FIGURA 12 – OPERADOR DE DOIS PONTOS



Fonte: Goldberg, Goldberg e Luna (2016).

- **Uniforme:** neste operador, em cada *locus* do filho, é sorteado, com distribuição de probabilidade uniforme (aleatoriamente), um dos pais para fornecer um alelo. Dois exemplos são destacados na Figura 13.

FIGURA 13 – OPERADOR UNIFORME



Fonte: Goldbarg, Goldbarg e Luna (2016)

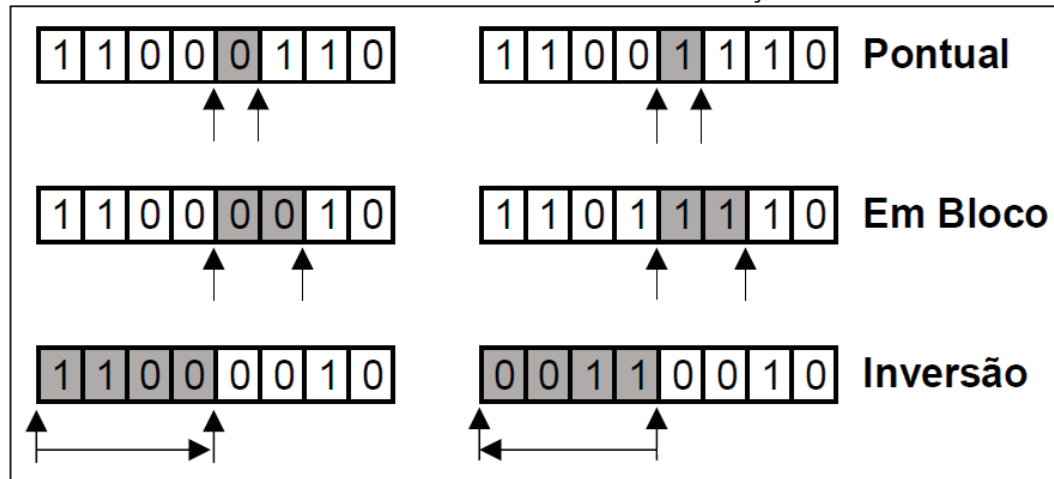
### 2.5.2 Operadores de Mutação

A mutação trata-se de um processo um processo estocástico de alteração do material genético dos cromossomos, em um mecanismo que produz novos indivíduos de maneira “assexuada”. Este tipo de operador pode ser aplicado sobre qualquer cromossomo da população, pais ou filhos, de acordo com a conveniência do algoritmo. A estratégia visa aumentar o poder de diversificação da população.

O primeiro elemento envolvido no processo de mutação seria a escolha dos cromossomos que serão alterados, o que sugere a possibilidade de existência de um mecanismo de seleção também para a mutação. Em seguida, por ser entendido como um processo estocástico, o processo leva em consideração uma probabilidade para o acontecimento da mutação, ou seja, nem todos os cromossomos selecionados precisam sofrer mutação, sendo a aptidão um indicador para a escolha do indivíduo a ser modificado. Por fim, é necessário ainda definir a extensão da mutação, isto é, o número de posições afetadas no cromossomo. Este último elemento pode ser implementado por meio de uma taxa de mutação, que representa o percentual de posições alteradas. No entanto, a orientação geral de que somente um alelo seja alterado por ocasião da mutação é um bom ponto de partida para o ajuste de algoritmos evolucionários.

Na Figura 14, é possível observar três tipos clássicos de operadores de mutação. O operador pontual corresponde à alteração de um alelo. No caso do operador em bloco, vários alelos são alterados. Por último, com o operador de inversão, os valores de um ou mais blocos são invertidos ou permutados.

FIGURA 14 – OPERADORES DE MUTAÇÃO



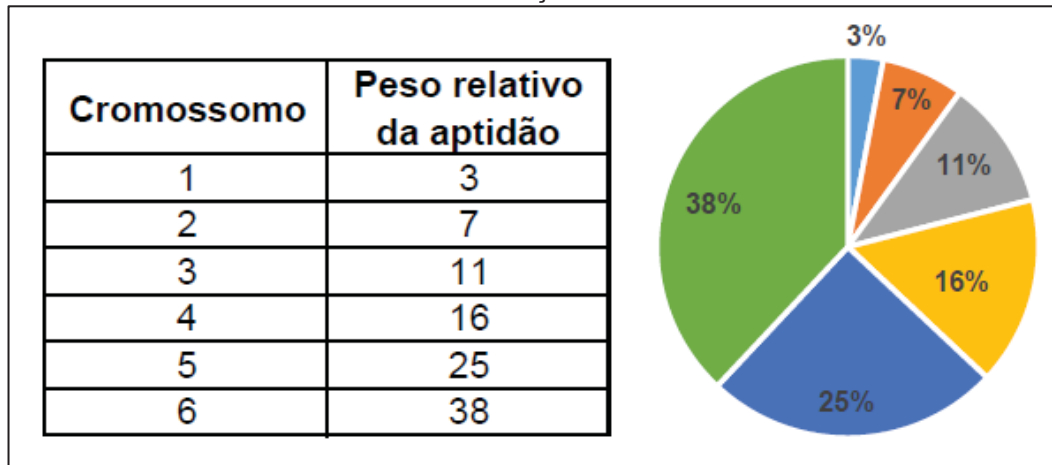
Fonte: Goldberg, Goldberg e Luna (2016)

### 2.5.3 Processos de Seleção

Um processo evolucionário artificial deve definir critérios que irão controlar a reprodução da população. Em princípio, a reprodução deverá ser uma oportunidade oferecida com maior chance aos cromossomos de melhor grau de aptidão. Por outro lado, se o tamanho da população for mantido constante, será necessário escolher critérios para remoção de cromossomos, elegendo na população formada pelos antigos pais e novos filhos quem sobreviverá e participará da próxima geração. Podem-se destacar, então, sete processos possíveis:

- **Torneio:** a seleção dos cromossomos ocorre por meio de uma espécie de competição, onde os vencedores serão selecionados para sobreviver. No torneio denominado binário, dois cromossomos são escolhidos aleatoriamente, avançando no torneio aquele que tiver a maior aptidão.
- **Classificação simples ou Elitismo:** os cromossomos são ordenados e escolhidos de acordo com a sua classificação. Por exemplo, por meio de um número  $k$  de vagas, as quais serão ocupadas pelos  $k$  cromossomos com maior aptidão ou por meio de um percentual.
- **Roleta:** cada cromossomo recebe uma área em uma roleta que é proporcional ao seu valor de aptidão. Em seguida, são feitos sorteios baseados nas aptidões, sem reposição, até que o número de cromossomos desejado seja escolhido (Figura 15).

FIGURA 15 – SELEÇÃO POR ROLETA



Fonte: Goldberg, Goldberg e Luna (2016)

- **Sorteio universal:** os cromossomos são sorteados com igual probabilidade, independente da aptidão.
- **Seleção dentro de agrupamentos:** os cromossomos são agrupados segundo certos critérios. Nos agrupamentos, aplica-se algum dos métodos de seleção aqui descritos.
- **Truncamento:** uma parcela da população é eliminada da competição. Na parcela restante, emprega-se um dos métodos de seleção aqui descritos.
- **Cotas ( $\mu$ ,  $\lambda$ ):** existem  $\mu$  vagas na população que devem ser preenchidas por  $\lambda$  candidatos selecionados segundo algum critério, e  $\lambda > \mu$ . A escolha dos  $\mu$  cromossomos no universo dos  $\lambda$  candidatos pode ser realizada por sorteio universal, pelo método da roleta ou por torneio.
- **Cotas híbridas:** uma parcela das vagas na população é preenchida por elitismo, enquanto as restantes, por qualquer método anteriormente descrito.

## 2.6 SIMULAÇÃO BIM 4D

Segundo Gledson e Greenwood (2016), o planejamento da construção é necessário para definir a duração dentro da qual o desempenho do projeto pode ser medido. O modelo de planejamento tradicionalmente adotado pelas empresas construtoras é bastante criticado pela sua incapacidade de levar em consideração aspectos espaciais e de localização quando as atividades são sequenciadas, como também pelo fato das programações e dos projetos serem feitos por softwares que geram arquivos de formatos distintos e também pelo fato da abordagem dos aspectos

espaciais ficar a critério apenas do profissional responsável pelo planejamento. Além disso, frequentemente o controle da produção durante a execução é priorizado em detrimento do seu planejamento em fases iniciais e a programação de atividades é enfatizada sem que se definam os métodos de produção.

Para Brito e Ferreira (2015), um dos entraves enfrentados pela indústria da construção é a dificuldade em visualizar corretamente o planejamento da obra no espaço, gerando cronogramas de interpretação abstrata para os usuários devido ao grande número de atividades e precedências. Modelos 4D, por sua vez, conectam intimamente aspectos espaciais e temporais no projeto, melhorando a confiabilidade nos cronogramas e minimizando problemas de comunicação.

Modelos em que a quarta dimensão de informação é o tempo, chamados de modelos 4D, são a combinação de modelos 3D com o planejamento da obra e são utilizados como fonte de planejamento visual da construção, além de gerarem um novo nível de visualização e entendimento dos processos por parte dos envolvidos no empreendimento (BIOTTO; FORMOSO; ISATTO, 2013).

O uso da modelagem 4D se apresenta como uma nova opção de ferramenta de auxílio à tomada de decisão na gestão da produção, pois as características de empreendimentos complexos demandam ações, métodos, técnicas e ferramentas apropriadas para gerenciá-los com sucesso (BIOTTO; FORMOSO; ISATTO, 2013).

A modelagem 4D pode ser feita utilizando modelos CAD e BIM. O planejamento da construção é conectado ao modelo 3D, permitindo a visualização da sequência construtiva e o cronograma do edifício. Ferramentas CAD 4D permitem ao modelador executar o planejamento visualmente e comunicar as atividades no contexto de espaço e tempo. Torna-se possível gerar animações 4D, que consistem em vídeos ou simulações virtuais do cronograma (BIOTTO; FORMOSO; ISATTO, 2013).

A funcionalidade do processo de planejamento pode ser melhorada de várias maneiras com o uso de tecnologia 4D, dentre as quais: a habilidade de importar informações de um modelo ou de um ambiente de dados compartilhados; melhorias na identificação de atividades por meio de simulações no modelo; e cálculo de durações com o uso de processos automatizados de extração de quantidades. Esses aprimoramentos permitem a produção de programações mais precisas e uma



comunicação mais eficaz de aspectos do planejamento (métodos de construção e sequências, direcionamento daqueles que recebem o planejamento ao local exato de trabalho e os impactos do movimento de recursos e da logística no canteiro). Atividades de risco também podem ser mais facilmente identificadas e destacadas (GLEDSON; GREENWOOD, 2016).

A modelagem 4D com a tecnologia BIM, por sua vez, refere-se a utilização de ferramentas de análise que incorporam os componentes BIM e informações sobre o método de construção para otimizar o sequenciamento de atividades. Essas ferramentas incorporam o espaço, a utilização dos recursos e informações de produtividade (BIOTTO; FORMOSO; ISATTO, 2013).

De acordo com Brito e Ferreira (2015), os principais benefícios da utilização de modelos 4D para o planejamento e controle são: a visualização de erros de sequenciamento, conflitos entre equipamentos e restrições físicas do canteiro, o apoio na conformação dos fluxos de trabalho e a definição logística dos equipamentos, demonstrando a capacidade das informações contidas nos modelos 4D para apoiar a tomada de decisões gerenciais.

Gledson e Greenwood (2016) destacam que a modelagem BIM 4D visa melhorar as técnicas de planejamento da construção por meio de funcionalidades como:

- Visualização das relações de tempo e espaço das atividades construtivas.
- Análise da programação da construção para avaliar sua implementação.
- Redução de erros por meio da simulação e validação do planejamento, melhorando simultaneamente a comunicação entre os agentes envolvidos.

Nos modelos 4D, a representação do avanço das atividades ao longo do tempo utiliza cores, as quais são essenciais para o melhor entendimento e realização de análises gerenciais. Segundo Brito e Ferreira (2015), um dos desafios a serem vencidos para permitir uma melhor utilização dessa tecnologia é a definição de padrões de representação dos elementos construtivos. A visualização de elementos 3D em simulações 4D pode ser influenciada por propriedades gráficas como cor, iluminação ou transparência e pelo nível de interação. Em sua pesquisa, os autores destacam entre as diretrizes para visualização e representação dos elementos

construtivos o uso de legendas com as cores da representação, diferenciação de luminosidade para distinção de componentes e vários níveis de transparência para melhor visualização do estado das atividades.

A capacidade de visualização das informações de planejamento e controle nos modelos BIM 4D é um dos aspectos fundamentais para uma implantação bem sucedida. A habilidade de comunicar informações sobre o avanço e discrepâncias em relação ao planejado com eficácia é estratégica para o gerenciamento de projetos, pois possibilita a tomada de ações corretivas em tempo hábil (BRITO; FERREIRA, 2015).

### 3. MÉTODO

Nesta sub-seção, é apresentada primeiramente a estratégia de pesquisa utilizada no trabalho, e em seguida a abordagem proposta da presente dissertação, que conta com três etapas: a modelagem do problema, o módulo de otimização, onde é implementado o algoritmo genético, e o módulo de simulação, onde as programações geradas pelo algoritmo são visualizadas por meio de simulação BIM 4D.

#### 3.1 ESTRATÉGIA DE PESQUISA

O planejamento da pesquisa em sua dimensão mais ampla, envolvendo tanto a diagramação quanto a previsão de análise e interpretação da coleta de dados, é chamado de delineamento. Tal processo considera, entre outros aspectos, o ambiente em que são coletados os dados e as formas de controle das variáveis envolvidas, além dos procedimentos técnicos para a coleta e análise de informações, servindo, então, como base para a classificação das pesquisas (GIL, 2002).

O experimento, por sua vez, representa o melhor exemplo de pesquisa científica. Essencialmente, a pesquisa experimental consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo e definir formas de controle e de observação dos efeitos que a variável produz no objeto. Trata-se, portanto, de uma pesquisa onde o pesquisador é um agente ativo, e não um observador passivo (GIL, 2002).

As pesquisas experimentais constituem o mais valioso instrumento disponível aos cientistas para testar hipóteses que estabelecem relações de causa e efeito entre as variáveis. Em virtude de suas possibilidades de controle, os experimentos oferecem garantia muito maior do que qualquer outro delineamento no sentido de provar a influência de uma variável sobre outras (GIL, 2002).

Na presente pesquisa, foram desenvolvidas duas variações do algoritmo genético para solucionar o Problema de Programação de Projetos com Restrições de Recursos (RCPSP) aplicado a projetos de habitação de interesse social. Os algoritmos foram implementados por meio da linguagem de programação *Python 2.7*, no sistema operacional Windows 8. A condução do estudo ocorre primeiramente pela avaliação da eficiência do algoritmo, que é feita por meio de numerosos experimentos

com instâncias de projetos presentes na base de *benchmark* para o RCPSP, a PSPLIB (KOLISCH; SPRECHER, 1996). A partir da validação, o algoritmo é aplicado a um projeto real. Os parâmetros do algoritmo poderão ser modificados, permitindo testar quais geram os melhores resultados (minimização da duração).

Em Pesquisa Operacional (PO), no entanto, estuda-se sistemas reais que são aplicados à realidade do mercado. Estes sistemas, por sua vez, caracterizam-se como um conjunto de processos que interagem e que recebem entradas, oferecendo resultados para algum propósito. Os estudos de PO fazem uso de um procedimento computacional no sentido de derivar soluções para um problema, porém, após a obtenção de possíveis soluções, ainda é necessário testar os modelos de processos e aprimorá-los conforme necessário, para que possam ser implementados em cenários reais. Assim, outra abordagem oportuna para a presente pesquisa é a de modelagem e simulação, que consiste no processo de criar e experimentar um sistema físico por meio de um modelo matemático computadorizado (HILLIER; LIEBERMAN, 2010; TURRIONI; MELLO; 2012).

Segundo Turrioni e Mello (2012), a condução de pesquisas por meio de modelagem e simulação de sistemas tem quatro propósitos principais: conhecer mais a fundo a forma de operação do sistema; desenvolver políticas e recursos para aperfeiçoar o desempenho do sistema; testar novos conceitos e/ou sistemas antes de implementá-los; e obter informações sem incomodar o sistema atual. A simulação computacional é a representação de um sistema real, através de um modelo, utilizando um computador, trazendo a vantagem de se poder visualizar esse sistema, implementar mudanças e responder questões do tipo “o que aconteceria se” (*what-if*), minimizando tempo e custos. Assim, o objetivo da simulação é estudar o comportamento do sistema, sem que seja necessário modificá-lo ou construí-lo fisicamente.

Desse modo, o modelo de sistema implementado na presente pesquisa consiste na rede de relações de precedência entre as atividades de projeto e seus respectivos requisitos de recursos, que servem como dados de entrada a geração de programações aperfeiçoadas por meio do algoritmo genético. O cronograma aprimorado obtido é, então, testado através de simulação BIM 4D, a fim de verificar se a sequência de atividades pode ser aplicada a situações reais. Portanto, a pesquisa, em um primeiro momento, tem caráter experimental, uma vez que se busca

desenvolver o algoritmo genético e melhorar seu desempenho por meio de testes na modificação dos seus parâmetros, evidenciando aqueles que fornecem os melhores resultados. Em seguida, é utilizada a estratégia de modelagem e simulação para investigar como a ferramenta implementada pode melhorar a organização das atividades de projeto, buscando a redução de sua duração.

### 3.2. MODELAGEM DO PROBLEMA

#### 3.2.1 Pressupostos para Programação

São considerados projetos de conjuntos habitacionais, compostos por um grande número de unidades residenciais com projeto padronizado. Os pressupostos que descrevem o problema são oriundos do Problema de Programação de Projetos com Restrições de Recursos (RCPSP), descrito na sub-seção 5.2, com algumas adaptações:

- Um projeto é composto por diversas atividades que podem ser representadas em formato atividade-no-nó (*activity-on-the-node*), que consiste em um gráfico acíclico onde os nós representam as atividades e os arcos representam as relações de precedência. Ao conjunto de atividades do projeto, são incluídas mais duas atividades fictícias, uma no início do projeto, sendo, portanto, predecessora direta ou indireta de todas as outras, e uma atividade final, representando a conclusão da construção.
- As relações de precedência entre as atividades são conhecidas e não podem ser modificadas.
- As atividades são ligadas por meio de relações de precedência final-início. Portanto, nenhuma atividade pode ser iniciada antes que todas as suas predecessoras tenham sido terminadas.
- Todas as informações conhecidas sobre o projeto (precedências, durações, requisitos de recursos) são de caráter determinístico (não se modificam com o tempo).
- Admite-se apenas um modo de execução para cada atividade.
- As atividades podem utilizar mais de um tipo de recurso.
- Considera-se todas as atividades sendo repetitivas, realizadas obrigatoriamente em todas as unidades habitacionais do projeto.
- O ritmo de execução das atividades é considerado constante.

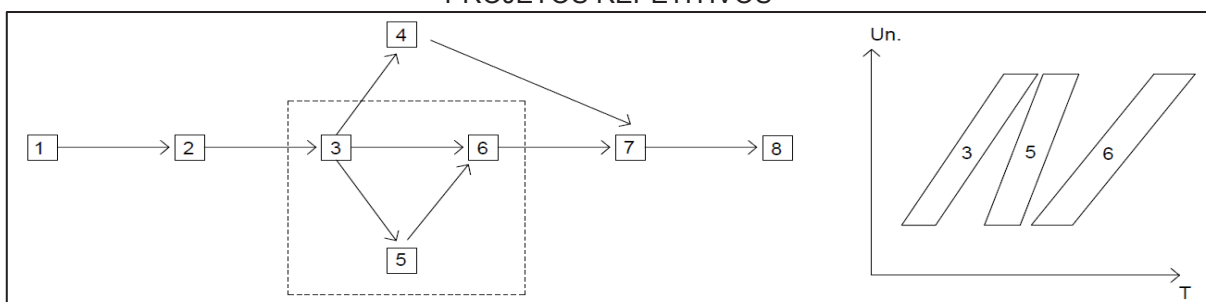
- A quantidade de trabalho em cada unidade habitacional não sofre variações de uma unidade para outra.
- Não é permitida a preempção, ou seja, a interrupção de uma atividade depois que ela é iniciada.
- Existe uma restrição de recursos a serem utilizados durante toda a duração do projeto, também de caráter determinístico.

### 3.2.2 Características do Projeto

De acordo com os pressupostos apresentados, a estruturação lógica do tipo de projeto abordado neste trabalho é obtida por meio de um gráfico orientado, onde os nós correspondem às atividades e os arcos representam as inter-relações entre as atividades. No entanto, como já evidenciado na sub-seção 5.1, esse tipo de representação não é o mais adequado quando se trata de projetos repetitivos, pois não é orientado para promover a continuidade das equipes entre as várias etapas de construção, além de não considerar o ritmo de produção e, quando aplicado a projetos lineares, gera problemas de visualização.

Assim, por meio da adaptação do método proposto por Ichihara (1999), o presente estudo buscará utilizar a técnica de gráfico orientado para modelar as inter-relações entre as atividades de uma unidade habitacional, em conjunto com um método de programação de projetos lineares, de modo a prever de maneira mais eficiente as relações final-início de cada pavimento (Figura 16). Assim, a interligação lógica absoluta fica garantida pelo gráfico, enquanto o segundo método considera a sincronização e a continuidade do trabalho das equipes. Segundo o autor, outra vantagem de utilizar uma abordagem baseada em um método de programação linear é a de que, assim, a programação das atividades repetitivas fica orientada pelos recursos e não relações de precedência.

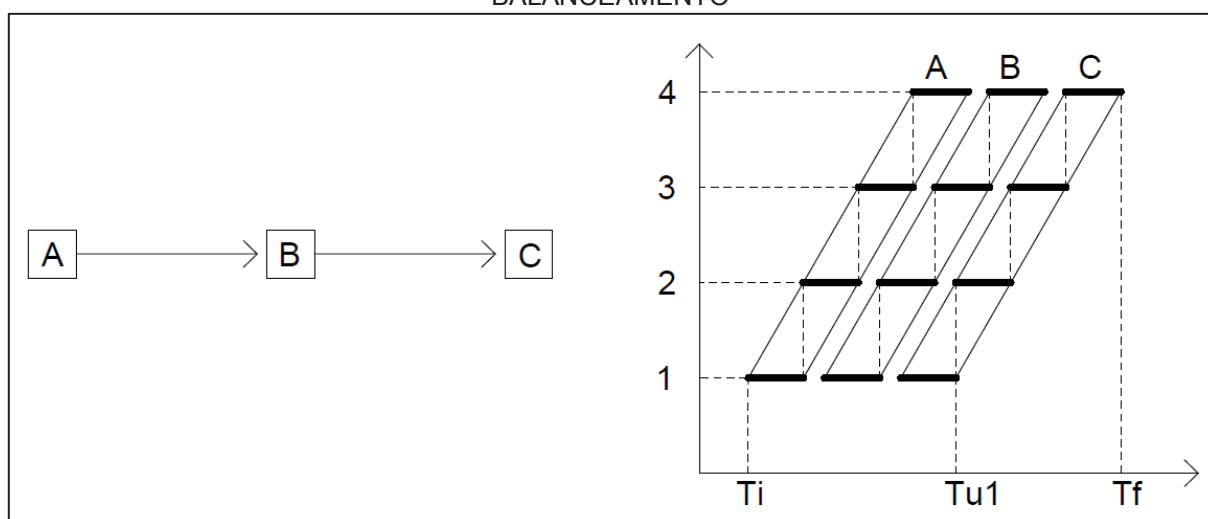
FIGURA 16 – INTER-RELAÇÕES ENTRE ATIVIDADES E MÉTODO DE PROGRAMAÇÃO DE PROJETOS REPETITIVOS



Fonte: Ichihara (1999).

Assim, o método de programação de projetos lineares adotado será a Linha de Balanceamento. Na Figura 17, é possível observar o funcionamento da técnica. Toma-se como exemplo um pequeno gráfico orientado composto por três atividades A, B e C, dispostas sequencialmente. À direita, é possível observar a linha de balanceamento correspondente ao projeto, com a repetição das atividades em quatro unidades habitacionais. Cada uma das barras horizontais de uma mesma atividade representa sua execução em um determinado espaço de tempo. O projeto se inicia no tempo  $T_i$ , que corresponde ao início dos trabalhos na primeira unidade habitacional. Os tempos  $T_{u1}$  e  $T_f$ , por sua vez, correspondem à data de finalização da primeira unidade e à data de conclusão da construção. É possível deduzir que cada equipe que trabalha em determinada atividade, desloca-se para a unidade seguinte após a conclusão dos trabalhos na unidade atual (ICHIHARA, 1999).

FIGURA 17 – REPRESENTAÇÃO DE UMA UNIDADE HABITACIONAL DO PROJETO EM ATIVIDADE-NO-NÓ E REPRESENTAÇÃO DA REPETIÇÃO POR MEIO DO MÉTODO DE LINHA DE BALANCEAMENTO



Fonte: adaptado de Ichihara (1999).

O método da Linha de Balanceamento adotado aqui levará em consideração a programação baseada em um tempo máximo de execução ( $D_t$ ). Por meio da diminuição do tempo de mobilização, ou seja, do tempo necessário para a conclusão dos serviços iniciais, e do tempo necessário para a execução de uma unidade repetitiva, é calculado então o tempo de ritmo ( $T_R$ ), que corresponde ao tempo máximo para a conclusão das atividades repetitivas:

$$T_R = D_T - (T_B + T_{MOB}) \quad (6)$$

Onde:

$T_R$  = Tempo de ritmo

$D_T$  = Duração total desejada para o projeto.

$T_B$  = Tempo base para a execução de uma unidade repetitiva.

$T_{MOB}$  = Tempo de mobilização, correspondente à conclusão dos serviços iniciais.

Em seguida, o ritmo de trabalho desejado pode ser calculado de acordo com a Equação 7 (MENDES JÚNIOR, 1999).

$$R = \frac{T_R}{(N - 1)} \quad (7)$$

Onde:

$R$  = Ritmo de trabalho desejado para o projeto.

$N$  = Número de unidades repetitivas.

Com o ritmo calculado, é possível então calcular a quantidade de equipes necessária para cada atividade (Equação 8) (MENDES JÚNIOR, 1999). Este cálculo resulta, em geral, em um número decimal, que deve ser corrigido para o número inteiro mais próximo.

$$N_E = \frac{D_U}{R} \quad (8)$$

Onde:

$N_E$  = Número de equipes.

$D_U$  = Duração para conclusão da atividade em uma unidade repetitiva.

Com a aproximação feita para o número de equipes, é possível então calcular o ritmo de execução específico de cada atividade, por meio da divisão da duração da tarefa pelo número de equipes corrigido (LIMMER, 1997).

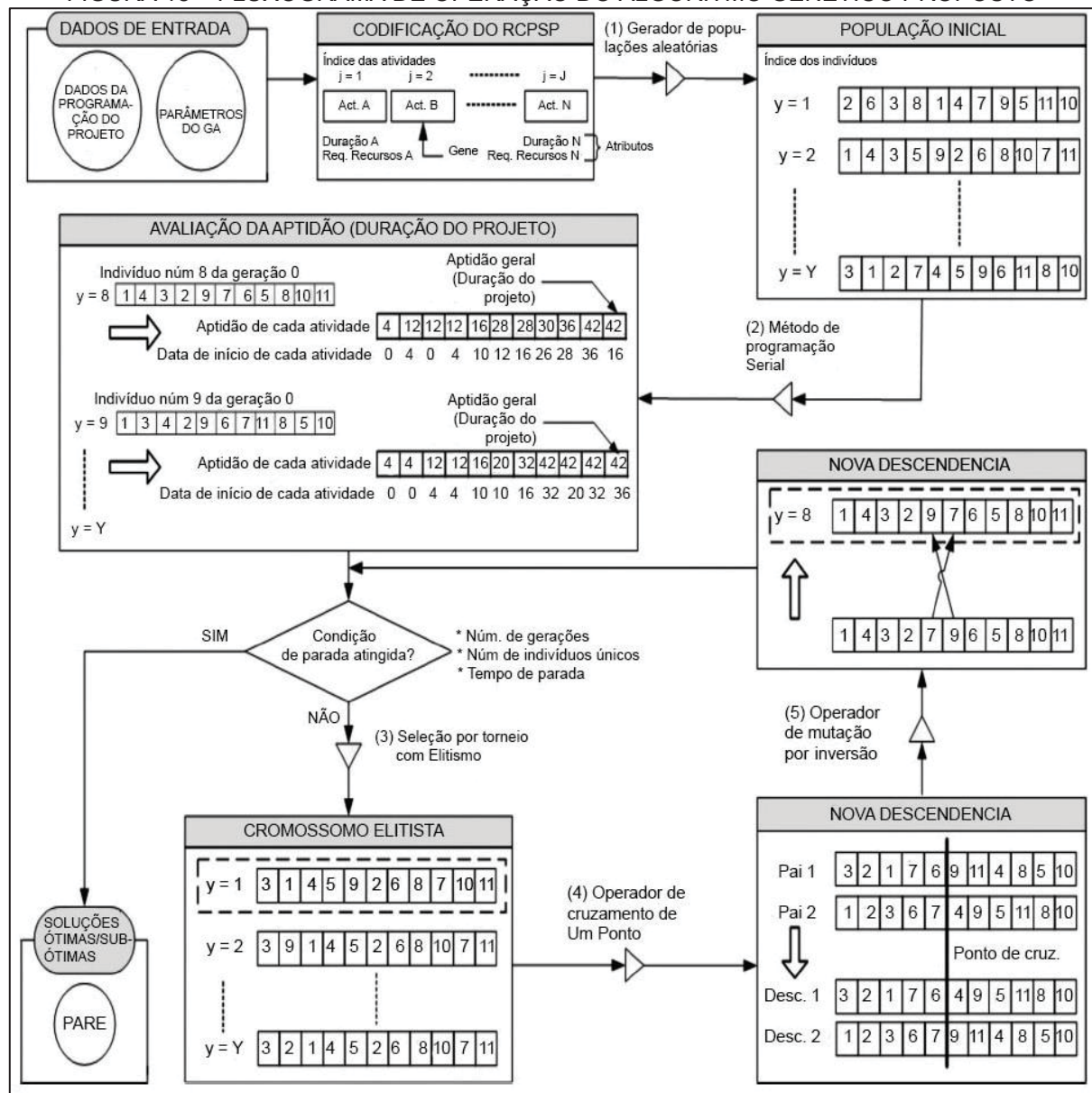
Admite-se também que, por se tratar de projetos de conjuntos habitacionais, as atividades possuem somente um sentido de execução, diferente do que acontece em projetos de edifícios, onde há atividades que começam no sentido térreo-cobertura, como a estrutura e a alvenaria, e atividades que são normalmente realmente no sentido contrário, como o revestimento externo.



### 3.3 MÓDULO DE OTIMIZAÇÃO

O módulo de otimização consiste na implementação do algoritmo genético, que é composto por diversas etapas, na seguinte ordem: 1. Geração da população inicial; 2. Avaliação da aptidão dos indivíduos; 3. Seleção dos indivíduos; 4. Aplicação do operador de cruzamento; e 5. Aplicação do operador de mutação (Figura 18).

FIGURA 18 – FLUXOGRAMA DE OPERAÇÃO DO ALGORITMO GENÉTICO PROPOSTO



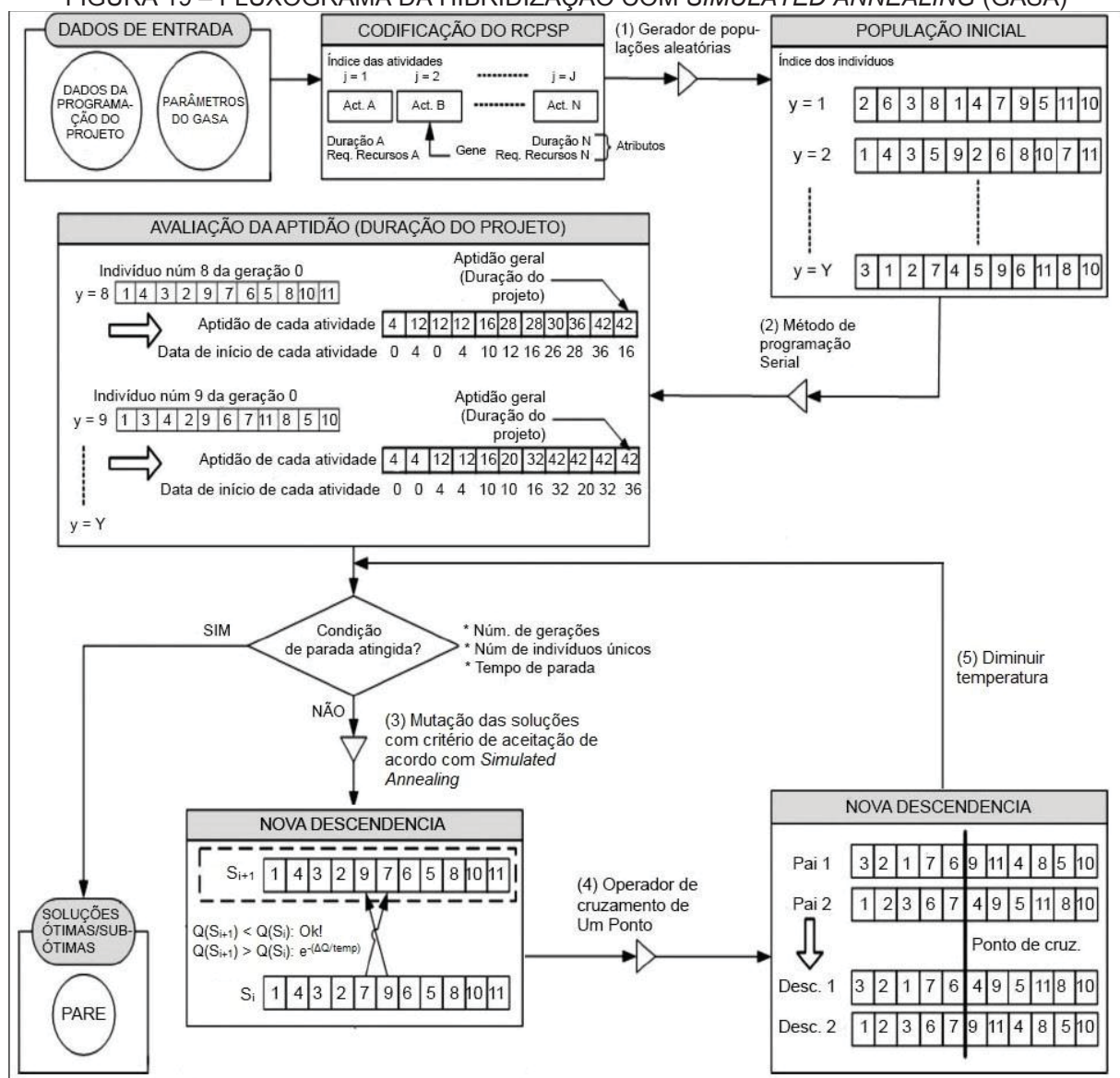
Fonte: adaptado de Kim e Ellis Jr. (2008)

No entanto, justamente pela grande possibilidade que o algoritmo genético oferece de ser modificado para tratar de diversos tipos de problema, é importante que se testem outras possibilidades de conduzir o processo evolucionário das soluções, para que assim se possa analisar melhor a influência dos operadores genéticos em

relação à sua capacidade de gerar novas soluções aprimoradas e também os desdobramentos da forma de aplicação destes operadores no tempo computacional.

Nesse sentido, será também implementado um algoritmo genético modificado, que utiliza o conceito do algoritmo *Simulated Annealing* para modificar as soluções da população inicial (Figura 19). O critério de aceite probabilístico do algoritmo será utilizado para aceitar ou não as modificações nos indivíduos da população inicial, e a temperatura será diminuída a cada iteração, de modo que, à medida que o algoritmo evolui, se torna menos provável que mutações piores sejam aceitas.

FIGURA 19 – FLUXOGRAMA DA HIBRIDIZAÇÃO COM *SIMULATED ANNEALING* (GASA)



Fonte: adaptado de Kim e Ellis Jr. (2008)

A maior diferença entre o GA tradicional implementado e essa hibridização com *Simulated Annealing* é a decisão sobre aceitação de soluções, uma vez que no

GA qualquer mutação é aceita independente de sua qualidade. Além disso, segundo Bettemir (2009), o uso de métodos complementares como o *Simulated Annealing* em conjunto com o GA pode aumentar significativamente a convergência do algoritmo, diminuindo o esforço computacional.

Apesar da implementação dos dois algoritmos ser ligeiramente diferente, os operadores usados nos dois são os mesmos. A seguir são, então, especificados os parâmetros implementados, assim como seus mecanismos de funcionamento.

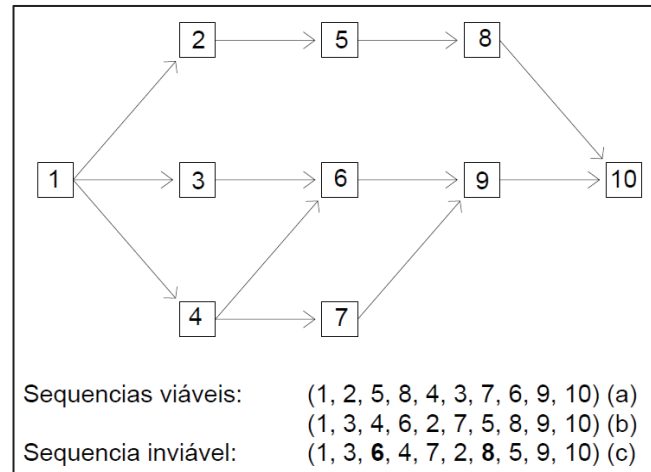
### 3.3.1 Codificação das Soluções e Geração da População Inicial

A codificação do problema de otimização demanda a criação de uma representação da programação do projeto que leve em consideração uma estrutura de prioridades entre as atividades. O algoritmo proposto no presente trabalho adota uma codificação baseada em permutação, caracterizada por uma Lista de Atividades  $\lambda = (j_1, j_2, j_3, \dots, j_n)$ , sugerida na literatura como uma representação relevante para a solução do RCPSP (KOLISCH; HARTMANN, 1999; KIM; ELLIS JR., 2008).

Assim, as soluções são aqui representadas em um cromossomo que corresponde a uma sequência de atividades. Cada gene no cromossomo, por sua vez, é caracterizado por um número que representa uma atividade. Uma tarefa tem uma prioridade mais baixa que todas as suas predecessoras e mais alta que as suas sucessoras. Desse modo, é garantido o atendimento às restrições de precedência, pois uma determinada atividade não pode aparecer na lista depois da posição onde se encontra alguma de suas sucessoras (KIM; ELLIS JR., 2008).

A representação pode ser exemplificada de acordo com a Figura 20, onde tem-se um gráfico representativo de um projeto e duas sequências possíveis *a* e *b*. No início do processo de sequenciamento em *a*, poderiam ser escolhidas as atividades 2, 3 ou 4, tendo a atividade 2 sido escolhida aleatoriamente. Em seguida, a decisão ficava entre as atividades 3, 4 ou 5, tendo sido escolhida a atividade 5, e assim por diante. A figura também mostra uma sequência inviável, onde as atividades 6 e 8 aparecem antes de suas predecessoras.

FIGURA 20 – SEQUENCIAMENTOS VIÁVEIS E INVIÁVEIS DE ACORDO COM AS RELAÇÕES DE PRECEDÊNCIA



Fonte: Ichihara (1999)

Desse modo, no ato da programação computacional, foi implementado um mecanismo que examina as relações de procedência entre as atividades na geração aleatória de indivíduos que compõem as populações iniciais formadas nas iterações do algoritmo. Serão atribuídos valores de prioridade à cada atividade, de modo que são sempre programadas primeiro as atividades com maiores valores, garantindo que estas apareçam primeiro na lista.

### 3.3.2 Mecanismo de Avaliação de Aptidão

A função-objetivo para o RCPSP, como evidenciado pela Equação 1, é a minimização da duração do projeto. Portanto, a aptidão de um indivíduo é representada pela data de finalização da última atividade do projeto. Assim, partindo de uma lista de atividades de um projeto e dispondo-se de suas relações de precedência e durações, é possível determinar as datas de término de cada atividade por meio de um procedimento heurístico de programação baseado em regras de prioridade. Para a presente abordagem, é adotado o Método Serial destacado na subseção 2.4.1.1.

Método Serial é feito por estágios, cujo número corresponde à quantidade de atividades no projeto. Associado a cada etapa, são construídos dois conjuntos: o conjunto de decisão e o conjunto de atividades programadas. As atividades já programadas fazem parte do segundo conjunto, pertencendo assim a uma espécie de programação parcial do projeto. O conjunto de decisão contém as atividades que ainda não foram programadas. Uma atividade é selecionada de acordo com a ordem da lista de atividades (representação do indivíduo) em cada estágio, e programada

para a data mais cedo que atenda às restrições de precedência e recursos, ou seja, a cada estágio é atribuído um valor de aptidão a uma atividade, representado pela sua data de finalização. Quando todas as atividades de um indivíduo já estão programadas, a sua *aptidão geral* é obtida pelo máximo valor entre a data de finalização da última atividade e o valor de aptidão da penúltima atividade (Figura 14) (KIM; ELLIS JR., 2008).

No entanto, a descrição acima se aplica a projetos cuja estrutura lógica pode ser organizada por redes do tipo PERT/CPM, ou seja, a data final de uma atividade  $j$  é determinada simplesmente somando a duração da atividade à data final de sua predecessora  $i$ . No entanto, o presente estudo trata de projetos repetitivos, sendo necessário levar em consideração métodos de planejamento que se adéquem a tais empreendimentos. Assim, é necessário destacar novas relações matemáticas para calcular as datas de início das atividades repetitivas, que levem em consideração o ritmo de execução de cada uma, como evidenciado na sub-seção 3.2.2.

Para isso, é necessário apresentar o conceito de *unidade crítica*, tratado em Ichihara (1999), onde o foco era edifícios com pavimentos-tipo, como *piso crítico*. Consiste na unidade que precisa ser verificada para programar a atividade  $j$ . Ao analisar esta unidade, torna-se redundante analisar as demais. Em outras palavras, a data de início da atividade  $j$  é a primeira data que torna possível preservar o sincronismo final-início na unidade crítica. Admite-se aqui que na construção de um conjunto habitacional as atividades possuem o mesmo sentido de execução, sendo possível identificar dois casos para o cálculo da data de início:

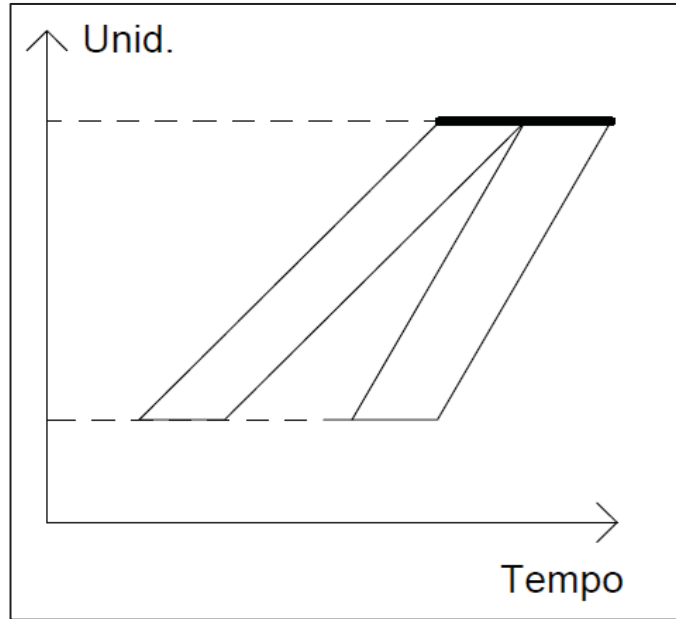
- **Caso 1:** O ritmo da atividade  $j$  é maior ou igual ao da atividade  $i$ .

A unidade crítica, nesse caso, é o último piso da atividade  $i$ , como demonstrado na Figura 21.

A data de início da atividade  $j$  pode ser calculada de acordo com a Equação 9.

$$\text{Início}_j = \text{Final}_i - \frac{(\text{UF}_j - \text{UI}_j)}{R_j} + 1 \quad (9)$$

FIGURA 21 – ANÁLISE DA UNIDADE CRÍTICA PARA O CASO 1



Fonte: Ichihara (1999)

Onde:

$\text{Início}_j$  = Data de início da atividade  $j$ .

$\text{Final}_i$  = Data de finalização da atividade  $i$ .

$\text{UI}_j$  = Unidade habitacional onde a atividade  $j$  é iniciada.

$\text{UF}_j$  = Unidade habitacional onde a atividade  $j$  é finalizada.

- **Caso 2:** O ritmo da atividade  $j$  é menor que o da atividade  $i$ .

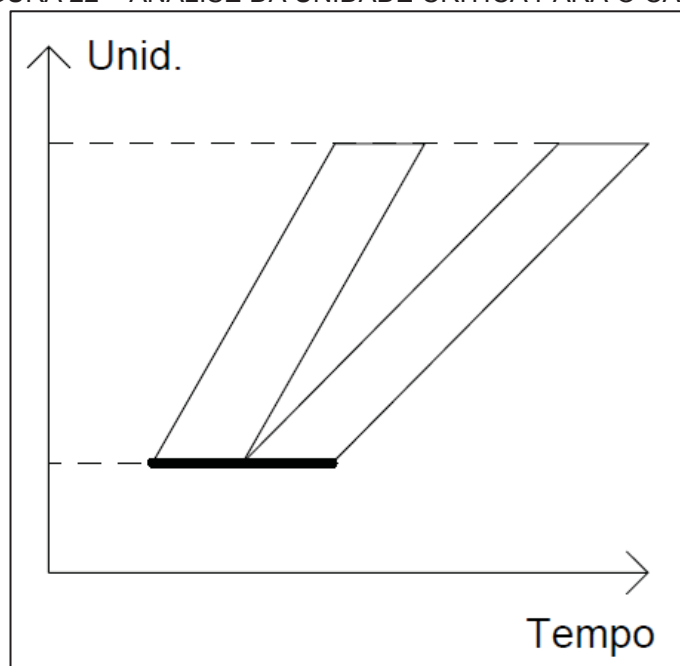
A unidade crítica, nesse caso, é a primeira unidade da atividade  $i$ , como visto na Figura 22.

A data de início da atividade  $j$  pode ser calculada, então, pela Equação 10.

$$\text{Início}_j = \text{Final}_i - \frac{(\text{UF}_i - \text{UI}_i)}{R_i} - \frac{(\text{UI}_i - \text{UI}_j)}{R_j} + 1 \quad (10)$$



FIGURA 22 – ANÁLISE DA UNIDADE CRÍTICA PARA O CASO 2



Fonte: Ichihara (1999)

Desse modo, a data de finalização de cada atividade  $j$  pode ser calculada normalmente somando sua data de início à duração total para a conclusão da atividade em todas as unidades habitacionais.

É importante ainda destacar que a análise da unidade crítica é feita somente pela avaliação do ritmo das atividades, identificando o grau de inclinação da linha de balanceamento. De acordo com a disponibilidade de recursos, no entanto, pode ser que as atividades não iniciem na primeira data após a conclusão de sua predecessora, sendo postergada para a primeira data onde haja recursos disponíveis.

### 3.3.3 Operadores de Seleção

É utilizada uma estratégia de seleção híbrida em cada algoritmo proposto. No algoritmo descrito na Figura 18, aqui chamada de GA, primeiramente será empregada uma seleção por Elitismo na população inicial, escolhendo uma porcentagem dos indivíduos com maiores valores de aptidão, que passarão obrigatoriamente para a próxima geração. A seleção por Elitismo permite ao algoritmo melhorar sua performance pois impede que se perca a melhor solução encontrada (KIM; ELLIS JR., 2008). Na implementação descrita na Figura 19, chamada GASA, as soluções serão modificadas por meio do operador de mutação, e os indivíduos originais poderão ou não ser substituídos por suas cópias modificadas, dependendo

do critério de aceitação proveniente do algoritmo *Simulated Annealing* (BETTEMIR, 2009).

Assim, no primeiro algoritmo, as melhores soluções são sempre passadas para as próximas gerações, enquanto no segundo tenta-se diversificar e melhorar as soluções de cada geração antes da etapa de cruzamento, de modo que seja mais provável que indivíduos com melhores valores de aptidão se recombinem.

Para o GASA (BETTEMIR; SONMEZ, 2013), é necessário estabelecer um método de definição da temperatura inicial do sistema, e também um esquema de resfriamento. Em Goldberg, Goldberg e Luna (2016), são apresentados diversos métodos para determinar a temperatura inicial, sendo a proposta de Van Laarhoven e Aarts (1987) o que mais se adequou a este trabalho:

$$T_0 = \frac{-\Delta E^+}{\ln \chi_0} \quad (11)$$

Onde  $\Delta E^+$  é a média aritmética, para um número aleatório de soluções, dos incrementos da função objetivo (aptidão). De modo geral, é necessário criar um número aleatório de indivíduos e determinar suas aptidões. Em seguida calcula-se a variação de uma para outra (da 1ª para a 2ª, da 2ª para a 3ª, etc.). Por fim, com todas as variações, obtém-se  $\Delta E^+$  pela média aritmética entre elas. Quanto a  $\chi_0$ , trata-se de um valor empírico no entorno de 0,8 (GOLDBARG; GOLDBARG; LUNA, 2016).

Quanto ao esquema de resfriamento, foi escolhida a proposta de Kirkpatrick et al. (1983) apud. Goldberg, Goldberg, Luna (2016):

$$T_i = \alpha T_{i-1} \quad (12)$$

Onde  $T_i$  é a nova temperatura,  $T_{i-1}$  a temperatura da iteração anterior e  $\alpha$  também um valor empírico que pode variar de 0.8 a 0.99.

Em seguida, para selecionar os indivíduos que passarão pelo operador de cruzamento, é adotada, em ambos os algoritmos, a estratégia de seleção por torneio, onde são formados dois pares de indivíduos por vez. Cada um dos pares vai, então, “competir” com base no seu valor de aptidão, sendo escolhidos os melhores cromossomos de cada par para passar pela etapa de cruzamento.



### 3.3.4 Operador de Cruzamento

O operador de cruzamento implementado nos algoritmos propostos foi o operador de Um Ponto. A escolha desse operador deve-se ao fato de que ele preserva partes dos cromossomos “pais”, mantendo as relações de precedência (HARTMANN, 1997). O processo é explicado em Kolisch e Hartmann (1999) da seguinte forma:

“Dadas duas atividades “pais”, uma “mãe”  $\lambda^M = [J_1^M, \dots, J_n^M]$  e um “pai”  $\lambda^P = [J_1^P, \dots, J_n^P]$ , a lista de atividades “filha”  $\lambda^C = [J_1^C, \dots, J_n^C]$  é definida da seguinte forma: depois de definir um número inteiro aleatório  $q$  tal que  $1 < q < n$ , as posições  $i = 1, \dots, q$  são tiradas da mãe  $\lambda^M$ , ou seja,  $J_i^C = J_i^M$ . As posições  $i = q + 1, \dots, n$  são, por sua vez, tiradas do pai  $\lambda^P$ . No entanto, os elementos que já haviam sido tirados da mãe não são considerados novamente. Obtem-se assim  $J_i^C = J_k^P$ , onde  $k$  é a atividade de menor índice dentre as atividades restantes na lista  $\lambda^P$  e que não estão na parte de  $\lambda^C$  tirada da mãe ( $J_k^P \notin \{J_1^C, \dots, J_i^C\}$ ).”

De maneira geral, o indivíduo gerado recebe integralmente uma parte da atividade “mãe” e, para a parte proveniente do “pai”, recebe os elementos que não figuram na parte “mãe”, nos mesmos índices que estavam no “pai”. As atividades que faltam são, então, postas na lista “filha” nos índices que ficaram sem atividades, seguindo a ordem em que figuram na lista “pai”. Kolisch e Hartmann (1999) exemplificam o processo com as seguintes listas, adotando  $q = 3$ :

$$\lambda^M = [1, 3, 2, 5, 4, 6]$$

$$\lambda^P = [2, 4, 6, 1, 3, 5]$$

$$\lambda^C = [1, 3, 2, 4, 6, 5]$$

Nesses exemplos é possível, então, notar que a lista “filha”  $\lambda^C$  recebe integralmente a primeira parte da lista “mãe”  $\lambda^M$  e apenas o elemento “5” da lista “pai”  $\lambda^P$ , pois os elementos 1 e 3 já estavam na primeira parte recebida. Os índices que ficaram sem elementos recebem então atividades da lista “pai” que ainda não figuram na lista “filha”, na ordem de seus índices (primeiro o elemento 4, depois o elemento 6). O processo é repetido em seguida tomando uma parte inteiramente do pai, gerando, no fim, dois novos indivíduos.

### 3.3.5. Operador de Mutação

É implementado nos algoritmos propostos o operador de mutação por Inversão. Em Klimek (2010) são apresentados alguns operadores de mutação, sendo adotado neste trabalho o operador de inversão de elementos adjacentes (*Swap Adjacent*). Para isso, antes de realizar a operação, o operador examina quais

elementos na lista podem ser invertidos com seus adjacentes sem violar as relações de precedência. Em seguida, um desses elementos é escolhido aleatoriamente para que a operação seja realizada.

#### 3.3.6. Critério de Parada

São utilizados três critérios de parada para algoritmo: número máximo de gerações, tempo limite de processamento e número de indivíduos únicos. É possível que vários indivíduos possuam o mesmo valor de aptidão, mesmo que as datas de início das atividades sejam completamente diferentes, como é possível observar na etapa de avaliação de aptidão na Figura 18. Assim, os indivíduos únicos, no espaço de busca, podem estar relacionados com a mesma programação, que representa a solução obtida para o RCPSP (KIM; ELLIS JR., 2008).

### 3.4 MÓDULO DE SIMULAÇÃO

No módulo de simulação, a execução da construção por meio das programações geradas pelo algoritmo genético poderá ser visualizada virtualmente. Esta etapa é importante pois as soluções determinadas pelo algoritmo são abstratas e não levam em consideração condições espaciais ou de construtibilidade. Assim, a partir da simulação 4D, será possível avaliar se as programações podem ser seguidas da maneira que foram criadas pelo algoritmo ou se são necessárias modificações suplementares. A simulação entra aqui como mais uma alternativa para a tomada de decisão. Assim, será possível decidir qual programação gerada pelo GA seria melhor de acordo com as prioridades do projeto e ainda resolver possíveis problemas identificados por meio da análise visual do modelo BIM.

A linguagem *Python 2.7* oferece a possibilidade de importar dados de listas para um arquivo no formato *.csv*. Essas informações podem, então, ser diretamente importadas para o *software* de simulação BIM 4D *Autodesk Navisworks*, que já vem com a opção de ler e interpretar dados de arquivos no formato *.csv*, sem a necessidade de instalar *plug-ins* ou de salvar os dados extraídos do módulo de otimização em algum outro formato. O projeto, por sua vez, será modelado no *software* BIM *Autodesk Revit*.

## 4. RESULTADOS

A seguir são apresentados os resultados dos experimentos realizados através dos dois algoritmos implementados e, em seguida, demonstrada a aplicação do módulo de simulação descrito na sub-seção 3.4.

### 4.1 EXPERIMENTOS COMPUTACIONAIS

Nesta seção, serão apresentados os resultados dos experimentos computacionais com os algoritmos implementados. Os testes foram feitos em instâncias de projeto tiradas da PSPLIB (KOLISCH; SPRECHER, 1996), uma base de *benchmark* específica para o RCPSP, disponível online. As instâncias são arquivos no formato .txt, pode ser lidos pela linguagem *Python 2.7*. Cada instância possui todos os dados de duração e requisitos de recursos de cada atividade, considerando 4 tipos de recurso diferentes para cada projeto. Também são fornecidas as relações de precedência e limitação de recursos a ser considerada. Um exemplo de como as informações encontram-se organizadas nos arquivos utilizados pode ser verificado no Apêndice D. Além das instâncias, o algoritmo também foi testado em três exemplos retirados dos estudos de Kim e Ellis Jr. (2008) e Bettemir e Sonmez (2013), por tratarem-se de trabalhos voltados para o setor de construção e também pelo fato dos autores terem encontrado soluções ótimas para os projetos utilizados, permitindo a melhor avaliação da eficiência do algoritmo.

Foram escolhidas aleatoriamente 15 instâncias com 30 atividades, 15 com 60 atividades e 15 com 90 atividades para os testes. Para as instâncias de 30 atividades, as soluções ótimas são conhecidas e fornecidas na PSPLIB. Para as instâncias com maior número de atividades, tem-se apenas valores de “limites inferiores” ou *lower bound*, que são os melhores valores encontrados para as instâncias até o momento. Para cada uma das 45 instâncias, foram realizadas dez rodadas, registrando-se a melhor solução encontrada e o tempo computacional despendido, totalizando 450 experimentos. A configuração utilizada para os parâmetros do algoritmo foi a seguinte:

- Tamanho da população: 50
- Taxa de recombinação: 0,5
- Tamanho do elitismo: 2 indivíduos
- Número máximo de gerações: 100

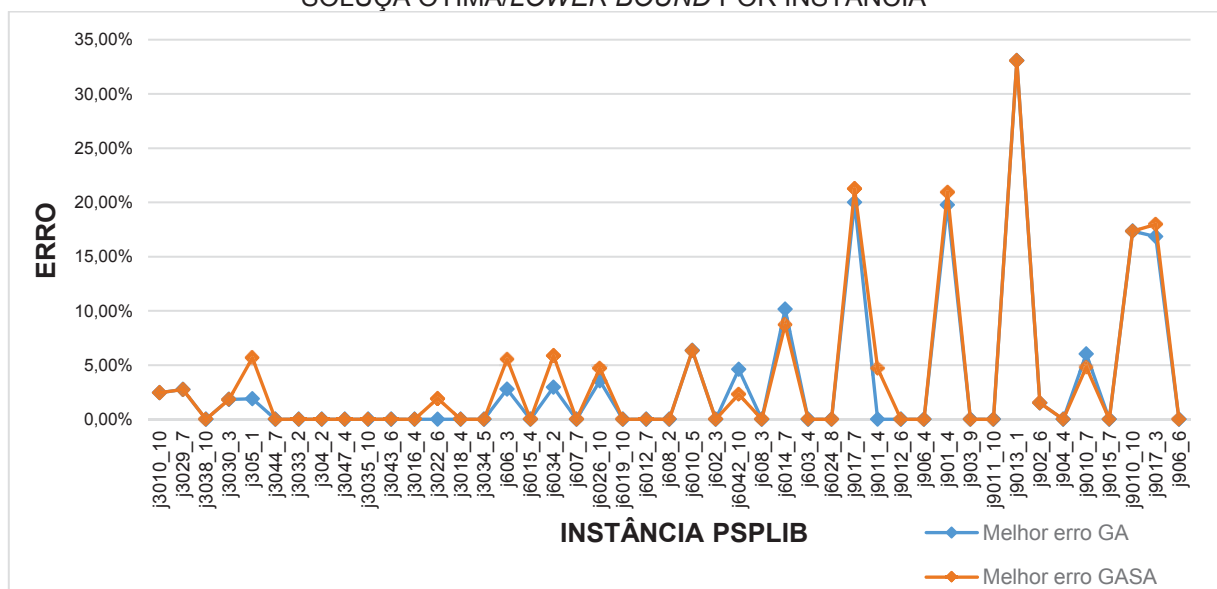
- Esquema de resfriamento  $\alpha$ : 0,8
- Número de indivíduos únicos: 25
- Tempo máximo: 4200 segundos

Para a taxa de mutação, foi adotada abordagem apresentada em Bäck e Fogel (2018), onde a esta é calculada pelo inverso do tamanho do indivíduo, ou seja, se o indivíduo é composto por 30 atividades, a taxa de mutação será  $1/30$  (0,033). Os demais parâmetros foram definidos de acordo com Kim e Ellis Jr. (2008), em razão dos bons resultados apresentados pelos autores.

O algoritmo e os operadores genéticos foram programados por meio da linguagem de programação *Python 2.7*. Os experimentos por sua vez, foram realizados em um computador Intel Core i5, 1,7 GHz, com 8GB de memória RAM. Os resultados dos experimentos são apresentados no Apêndice B.

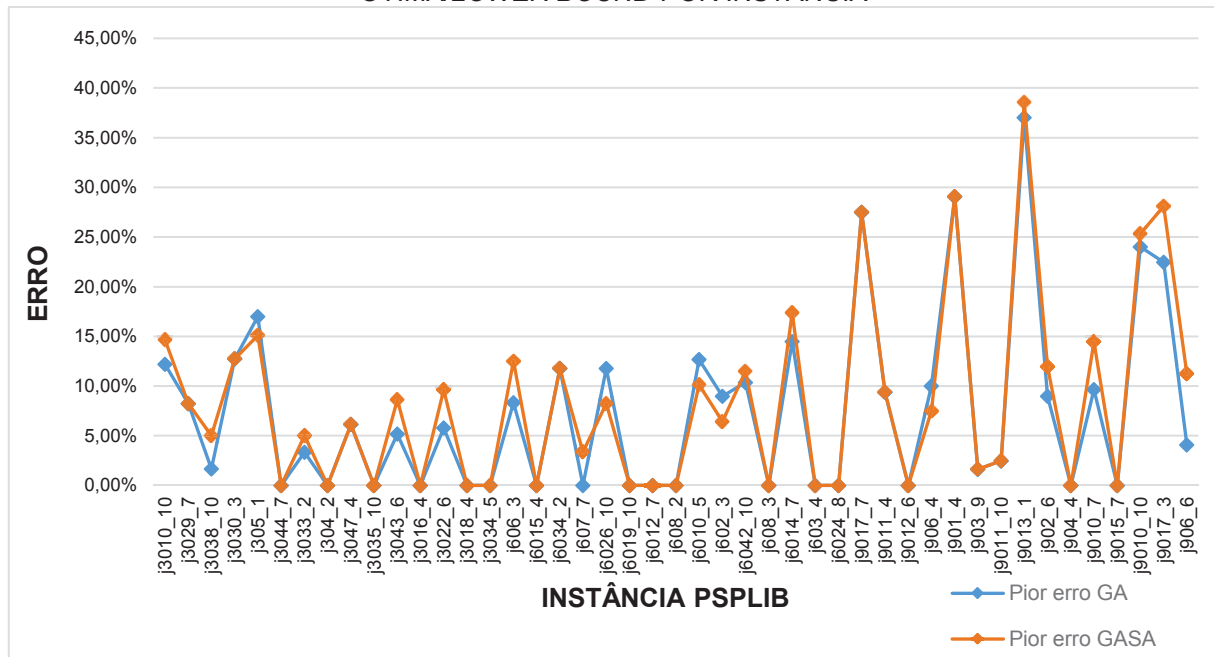
Observou-se com os experimentos que os dois algoritmos são capazes de gerar soluções melhores ao longo das gerações. O GA, no entanto, teve um desempenho ligeiramente melhor, como é possível observar nos Gráficos 1 e 2, que mostram a porcentagem de diferença entre a melhor e a pior solução encontradas nas dez rodadas e a solução ótima ou *lower bound* informado na PSPLIB.

GRÁFICO 2 – DIFERENÇA PERCENTUAL ENTRE A MELHOR SOLUÇÃO ENCONTRADA E A SOLUÇÃO ÓTIMA/LOWER BOUND POR INSTÂNCIA



Fonte: o autor, 2019

GRÁFICO 3 – DIFERENÇA PERCENTUAL ENTRE PIOR SOLUÇÃO ENCONTRADA E A SOLUÇÃO ÓTIMA/LOWER BOUND POR INSTÂNCIA

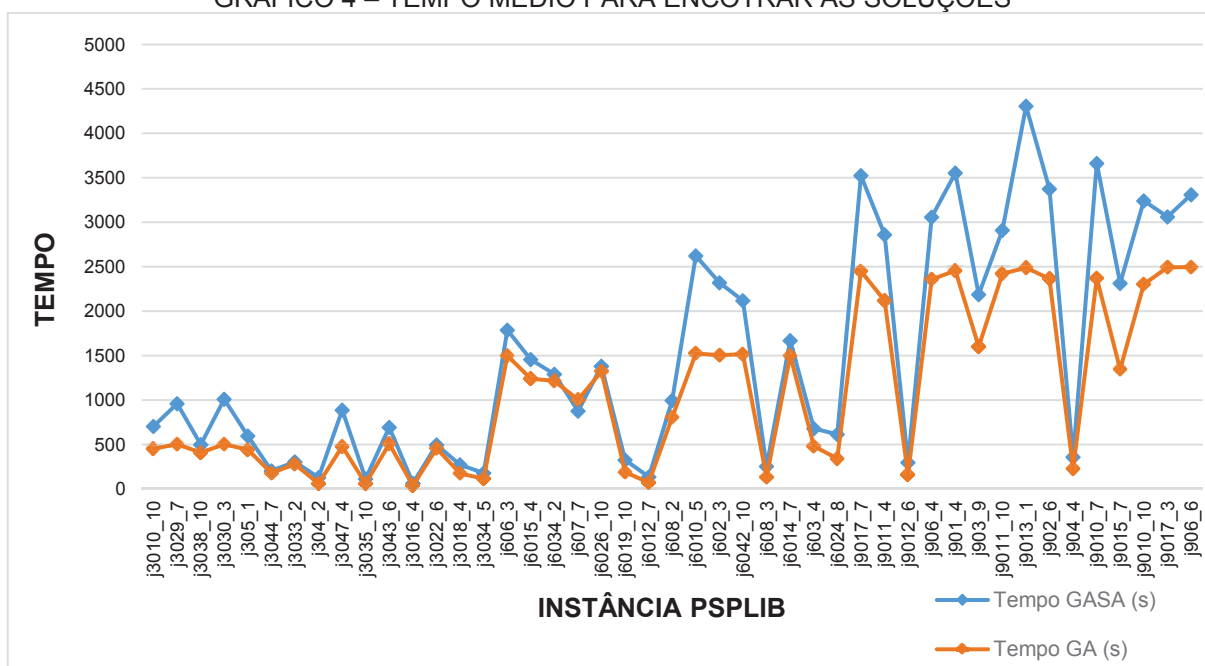


Fonte: o autor, 2019

Para as instâncias de 30 atividades, como esperado, foram encontradas soluções próximas das ótimas mais facilmente. No entanto, os algoritmos implementados foram capazes de encontrar os valores ótimos para a maioria das instâncias, com um pior desempenho para as instâncias de 90 atividades.

Como é possível observar no Gráfico 4, à medida que o tamanho dos problemas aumenta, também cresce a dificuldade de encontrar boas soluções e o tempo computacional necessário. O GASA teve um desempenho pior que o GA também quanto ao tempo computacional, o que se deve ao fato de que, na inicialização do algoritmo, além de calcular a aptidão das soluções da população inicial, ele ainda calcula a aptidão das novas soluções que possivelmente as substituirão.

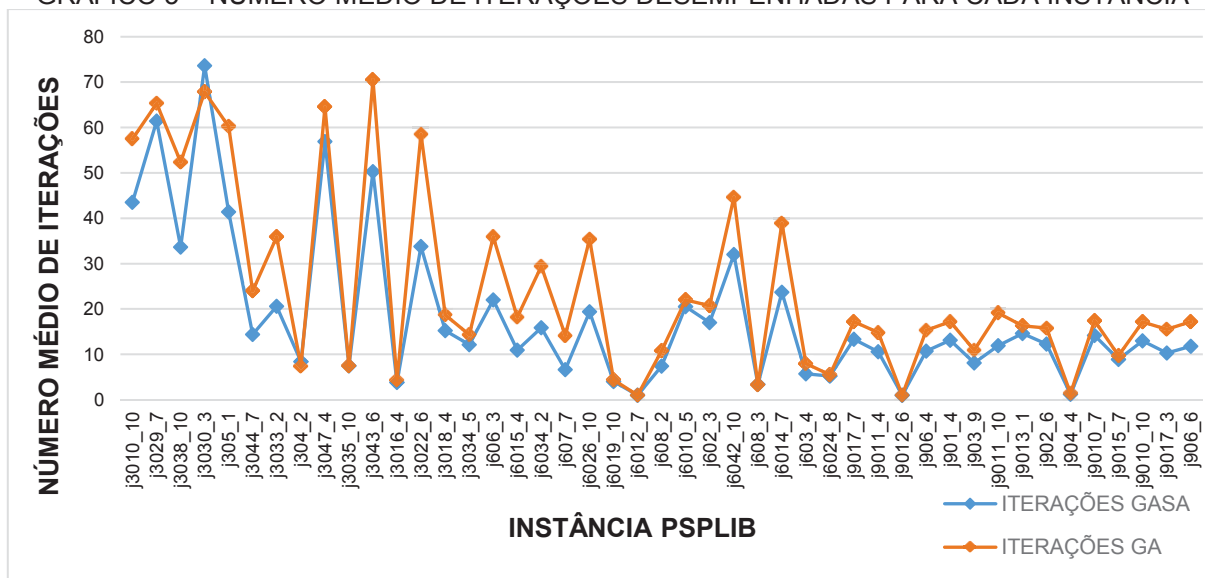
GRÁFICO 4 – TEMPO MÉDIO PARA ENCONTRAR AS SOLUÇÕES



Fonte: o autor, 2019

No entanto, foi possível observar também que o GASA implementado atendeu ao pressuposto verificado em Bettemir (2009), de que a hibridização com *Simulated Annealing* aumenta a capacidade de convergência do algoritmo para boas soluções. De fato, o GASA foi capaz de melhorar as soluções em um número de gerações (iterações) menor que o GA para praticamente todas as instâncias, como é possível observar no Gráfico 5.

GRÁFICO 5 – NÚMERO MÉDIO DE ITERAÇÕES DESEMPENHADAS PARA CADA INSTÂNCIA

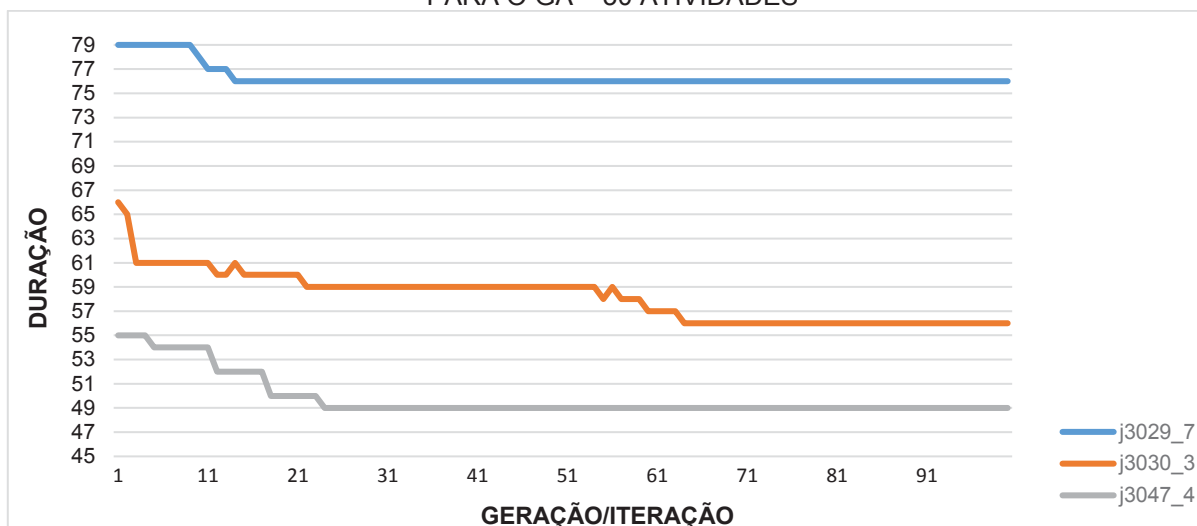


Fonte: o autor, 2019

Foram escolhidos ainda três grupos de três instâncias, cada um com um número de atividades, para verificar o comportamento dos algoritmos até última geração, a fim de avaliar melhor tanto suas eficiências quanto a capacidade de melhoria das soluções ao longo das gerações. Para esta etapa, foram escolhidas instâncias que tiveram um comportamento de convergência mais lento. Os resultados podem ser vistos nos gráficos a seguir (6 a 11).

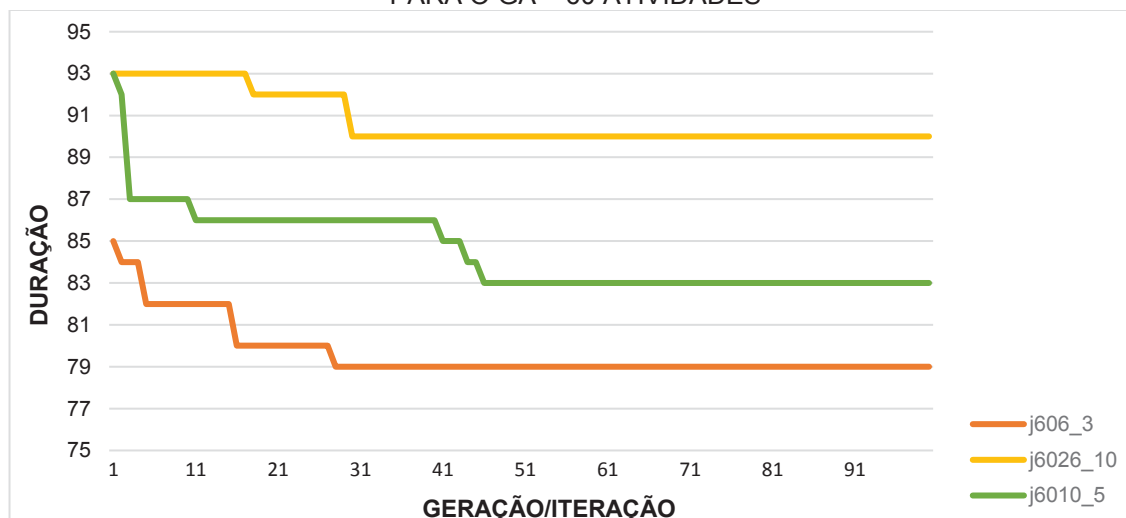
Nos gráficos 6 a 8, mostra-se o comportamento das instâncias j3029\_7, j3030\_3, j3047\_4, j606\_3, j6026\_10, j6010\_5, j9011\_4, j906\_4 e j9011\_10, quando aplicadas ao GA. Para todas instâncias, foi possível observar a capacidade do algoritmo de melhorar as soluções ao longo das gerações.

GRÁFICO 6 – COMPORTAMENTO DA MELHORIA DAS SOLUÇÕES AO LONGO DAS GERAÇÕES PARA O GA – 30 ATIVIDADES



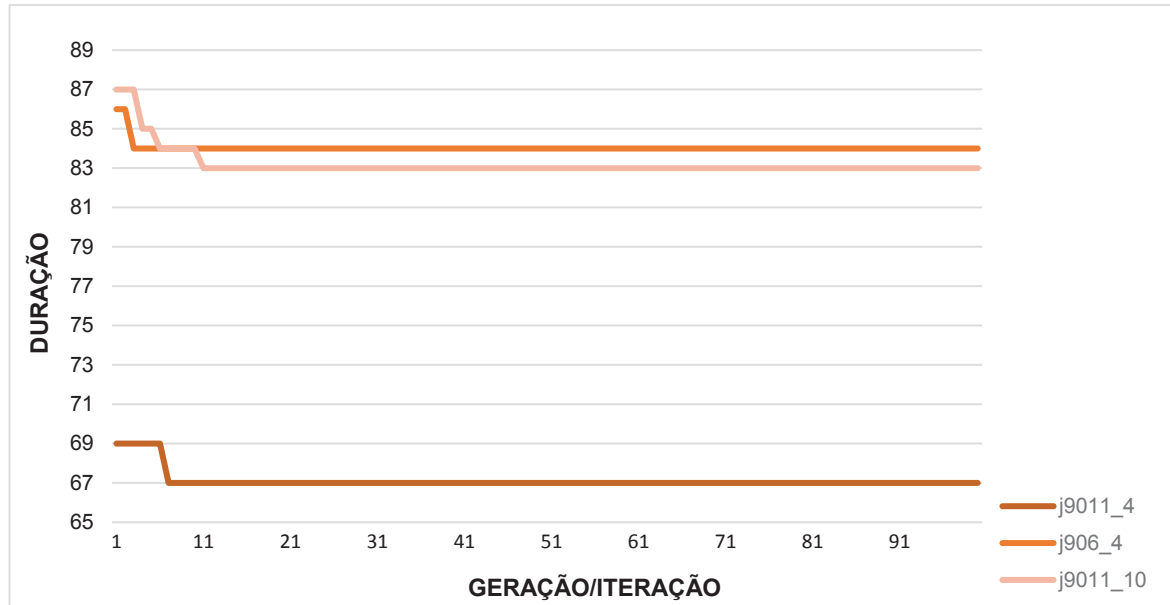
Fonte: o autor, 2019

GRÁFICO 7 – COMPORTAMENTO DA MELHORIA DAS SOLUÇÕES AO LONGO DAS GERAÇÕES PARA O GA – 60 ATIVIDADES



Fonte: o autor, 2019

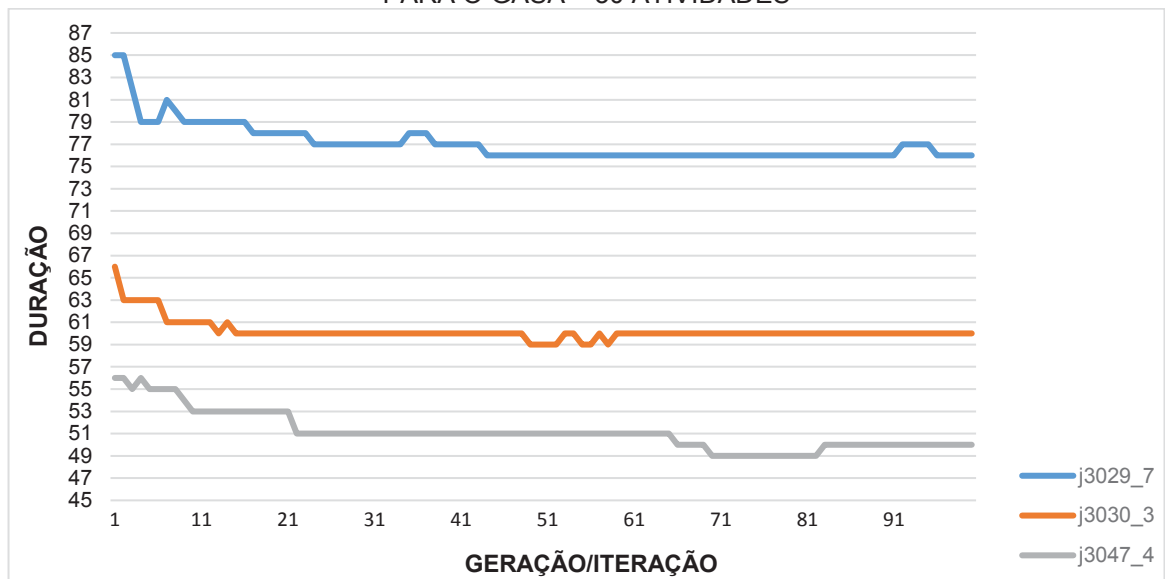
GRÁFICO 8 – COMPORTAMENTO DA MELHORIA DAS SOLUÇÕES AO LONGO DAS GERAÇÕES PARA O GA – 90 ATIVIDADES



Fonte: o autor, 2019

Por sua vez, os gráficos 9 a 11 mostram os comportamentos das mesmas instâncias quando aplicadas ao GASA, evidenciando, porém, um desempenho bem diferente em relação ao GA.

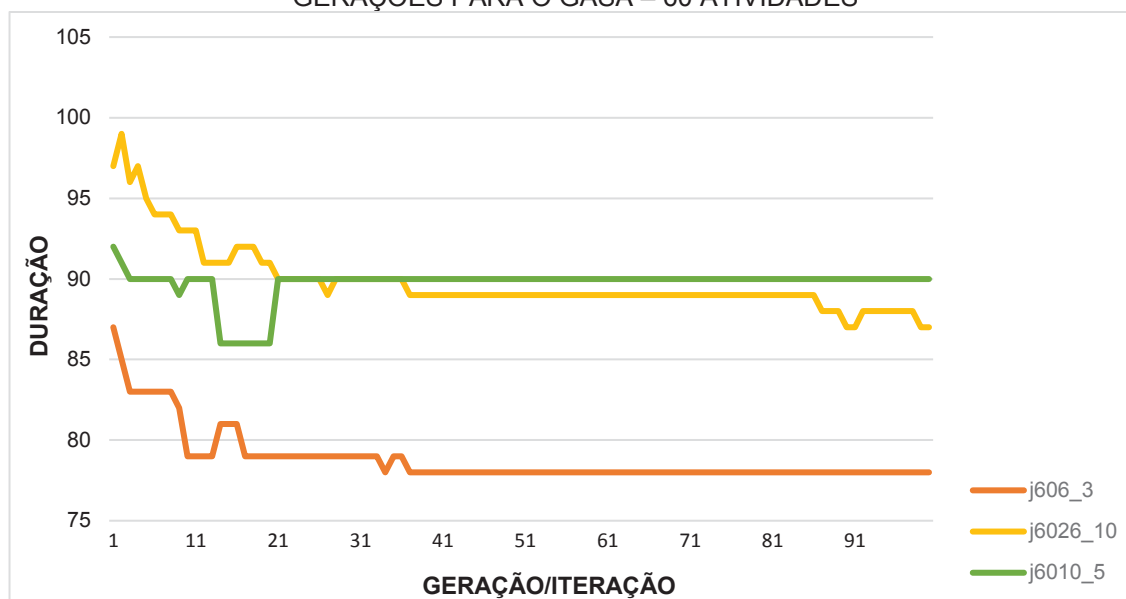
GRÁFICO 9 – COMPORTAMENTO DA MELHORIA DAS SOLUÇÕES AO LONGO DAS GERAÇÕES PARA O GASA – 30 ATIVIDADES



Fonte: o autor, 2019

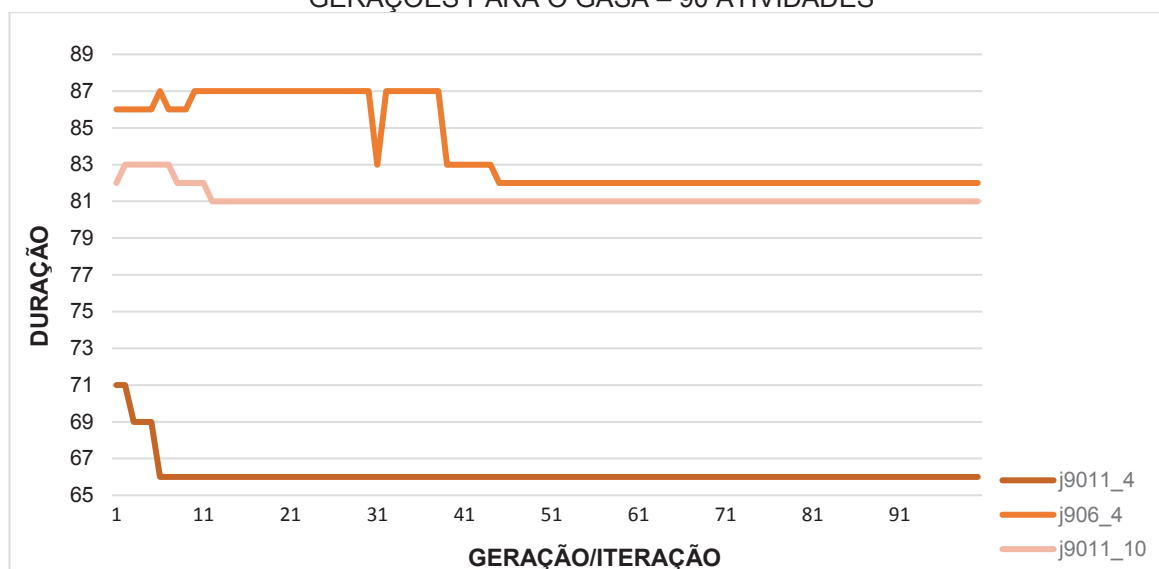


GRÁFICO 10 – COMPORTAMENTO DA MELHORIA DAS SOLUÇÕES AO LONGO DAS GERAÇÕES PARA O GASA – 60 ATIVIDADES



Fonte: o autor, 2019

GRÁFICO 11 – COMPORTAMENTO DA MELHORIA DAS SOLUÇÕES AO LONGO DAS GERAÇÕES PARA O GASA – 90 ATIVIDADES



Fonte: o autor, 2019

A partir da observação dos gráficos, foi possível observar mais uma vez a capacidade dos algoritmos de melhorar as soluções ao longo das iterações, demonstrando a implementação correta do procedimento evolucionário descrito na literatura. No entanto, observou-se que o GASA oferece um risco maior que o GA de perder boas soluções durante as iterações, sobretudo em fases iniciais. Isso se deve ao fato de que nas primeiras gerações, a temperatura do sistema ainda não foi muito reduzida, permitindo a aceitação de mutações piores. Ao contrário do GA, o GASA implementado não utilizou a técnica de Elitismo, que garante que as melhores

soluções sempre passem para a próxima geração. Este fato é muito claro pela análise dos Gráficos 6, 7 e 8, onde a tendência é sempre de reduzir as durações, enquanto o comportamento do GASA se mostra bem mais inconstante. No gráfico 10 é inclusive possível verificar a perda de uma solução muito melhor que a solução final encontrada para a instância j6010\_5, com uma diferença de 4 dias. Foi possível concluir também que, a partir da 50ª geração em média, os algoritmos convergem para ótimos locais, não sendo capazes de melhorar muito mais as soluções, o que justifica o uso de um tempo máximo de parada, como forma de poupar esforço computacional desnecessário.

Desse modo, é possível afirmar que o GA implementado é mais eficiente que o GASA, uma vez que demanda menos tempo para encontrar boas soluções e oferece menos risco de perdê-las ao longo das iterações. Por esse motivo, após a fase de experimentos, o GA foi aplicado a um projeto de um conjunto habitacional nos moldes do programa “Minha Casa Minha Vida” e o cronograma gerado pôde ser simulado no *Autodesk Naviswork*, como demonstrado a seguir.

#### 4.2 EXEMPLO DE APLICAÇÃO DO MÓDULO DE SIMULAÇÃO

Para demonstrar o funcionamento da fase de simulação, primeiramente o algoritmo foi testado com um projeto de um edifício no padrão do programa Minha Casa Minha Vida. Para isso, foi primeiramente elaborada uma rede de precedências para o projeto, assim como definidas as durações de cada atividade, como é possível observar na Tabela 2.

Tabela 2 – Duração e relações de precedência do primeiro projeto simulado (continua).

| Número da atividade | Nome da atividade     | Duração | Sucessoras |
|---------------------|-----------------------|---------|------------|
| 1                   | Início                | 0       | 2          |
| 2                   | Serviços preliminares | 10      | 3          |
| 3                   | Escavação             | 22      | 4          |
| 4                   | Fundações             | 16      | 5          |
| 5                   | Impermeabilização     | 9       | 6          |
| 6                   | Laje Térreo           | 10      | 7          |
| 7                   | Pilares Térreo        | 15      | 8          |
| 8                   | Vigas Térreo          | 15      | 9, 10      |
| 9                   | Alvenaria Térreo      | 25      | 26         |
| 10                  | Laje 1º pav.          | 10      | 11         |
| 11                  | Pilares 1º pav.       | 15      | 12         |
| 12                  | Vigas 1º pav.         | 15      | 13, 14     |

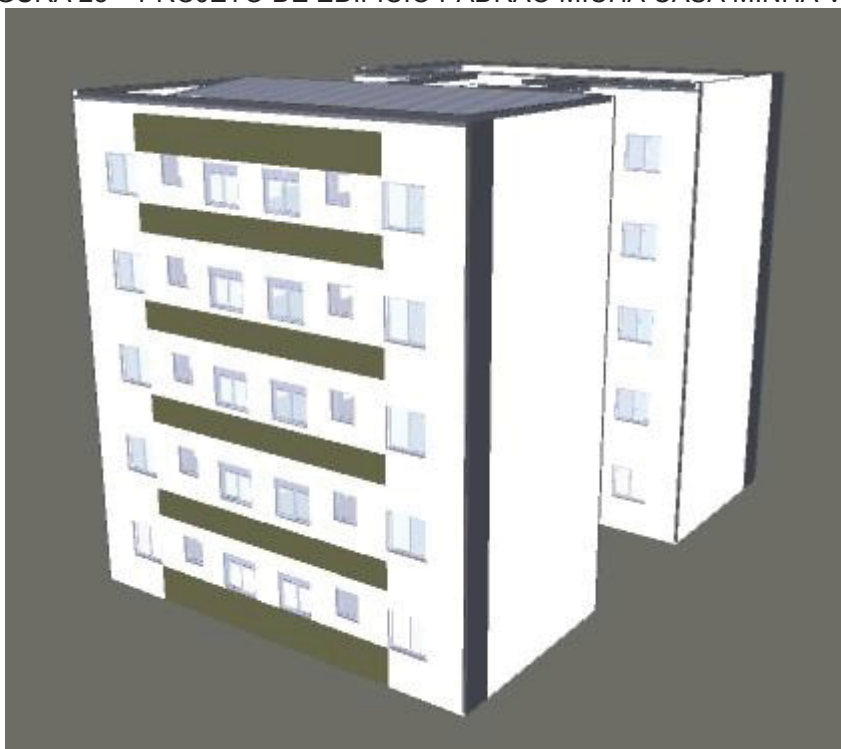
Fonte: o autor, 2019

Tabela 2 – Duração e relações de precedência do primeiro projeto simulado (continuação).

|    |                   |    |        |
|----|-------------------|----|--------|
| 13 | Alvenaria 1º pav. | 25 | 26     |
| 14 | Laje 2º pav.      | 10 | 15     |
| 15 | Pilares 2º pav.   | 15 | 16     |
| 16 | Vigas 2º pav.     | 15 | 17, 18 |
| 17 | Alvenaria 2º pav. | 25 | 26     |
| 18 | Laje 3º pav.      | 10 | 19     |
| 19 | Pilares 3º pav.   | 15 | 20     |
| 20 | Vigas 3º pav.     | 15 | 21, 22 |
| 21 | Alvenaria 3º pav. | 25 | 26     |
| 22 | Laje 4º pav.      | 10 | 23     |
| 23 | Pilares 4º pav.   | 15 | 24     |
| 24 | Vigas 4º pav.     | 15 | 25     |
| 25 | Alvenaria 4º pav. | 25 | 26     |
| 26 | Pintura           | 60 | 27     |
| 27 | Revestimento      | 80 | 29     |
| 28 | Cobertura         | 15 | 29     |
| 29 | Serviços finais   | 10 | 30     |
| 30 | Fim               | 0  |        |

Fonte: o autor, 2019

FIGURA 23 – PROJETO DE EDIFÍCIO PADRÃO MICHA CASA MINHA VIDA

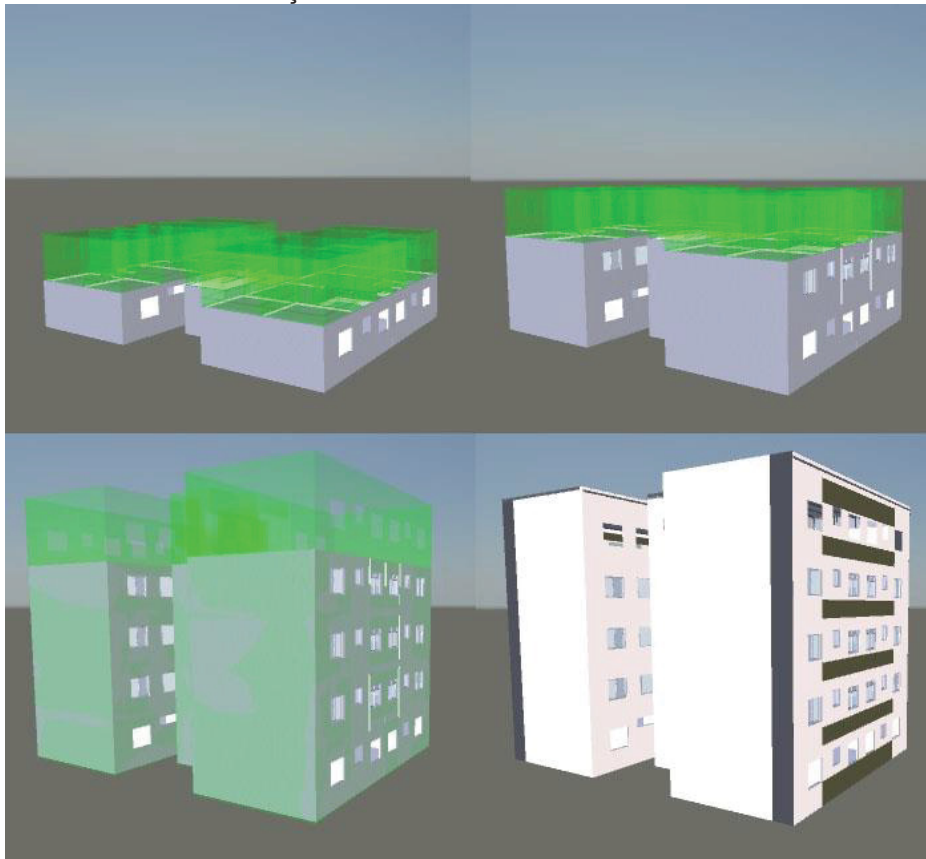


Fonte: o autor, 2019

Nesta primeira etapa, apesar das atividades repetitivas, a rede de precedências do projeto em questão foi utilizada deste modo para testar se o arquivo

.csv gerado pelo algoritmo poderia ser lido pelo programa *Autodesk Navisworks* e se o cronograma resultante seguiu uma sequência lógica, o que pôde ser confirmado pela simulação mostrada na Figura 23. Os parâmetros utilizados foram iguais aos dos experimentos computacionais realizados.

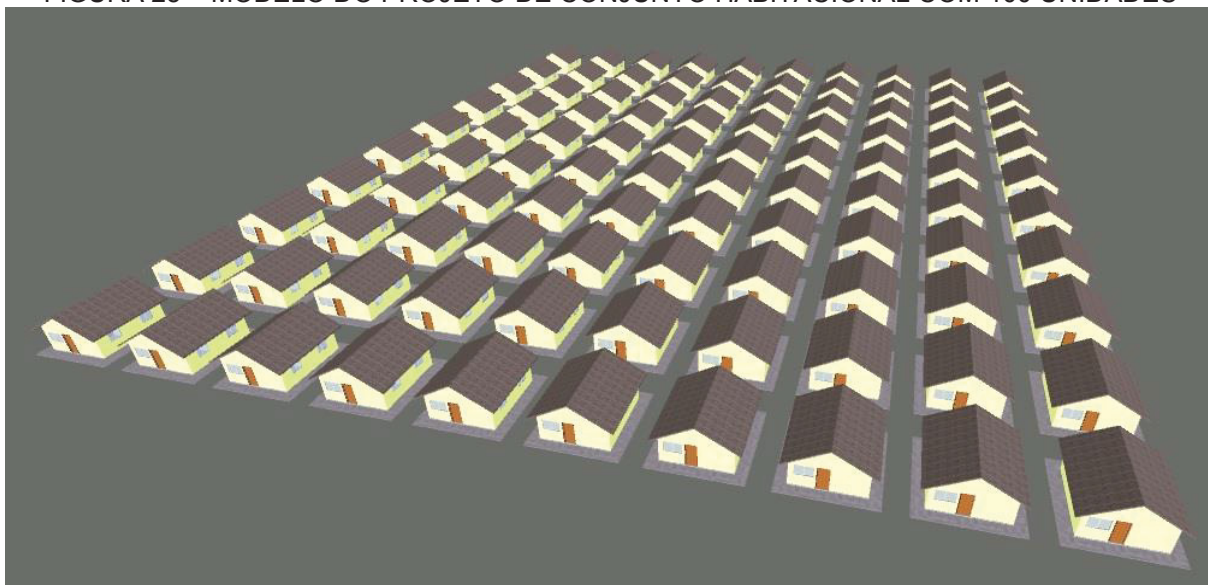
FIGURA 24 – SIMULAÇÃO DO PROJETO DE EDIFÍCIO NO NAVISWORKS



Fonte: o autor, 2019

Em seguida, como forma de exemplificar como os cálculos da Linha de Balanceamento apresentados na sub-seção 3.3.2 podem ser aplicados no processo de otimização, foi usado o projeto de habitação popular presente no Apêndice C. Para fins de demonstração, o modelo usado nesta etapa contou com 100 casas (Figura 24).

FIGURA 25 – MODELO DO PROJETO DE CONJUNTO HABITACIONAL COM 100 UNIDADES



Fonte: o autor, 2019

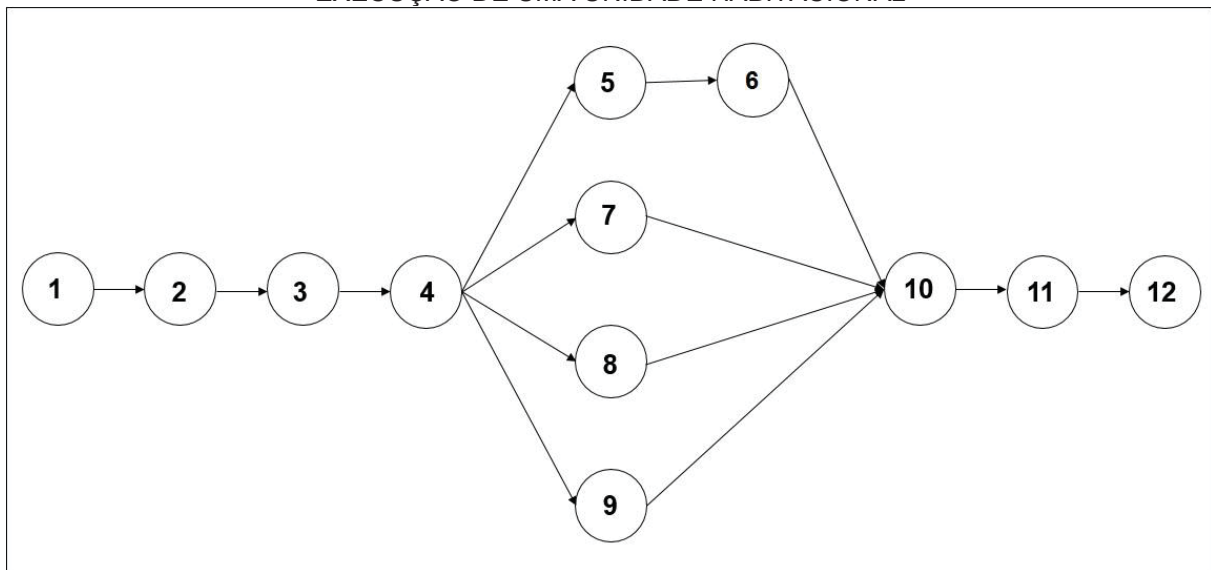
Foi elaborada também uma rede de precedências para uma unidade habitacional, assim como é feito ao aplicar o método da Linha de Balanceamento manualmente (Figura 24). Também foram determinadas as durações para cada atividade, como pode ser visto na Tabela 2.

Tabela 3 – Duração e relações de precedência do projeto considerado.

| Número da atividade | Nome da atividade       | Duração | Sucessoras |
|---------------------|-------------------------|---------|------------|
| 1                   | Início                  | 0       | 2          |
| 2                   | Viga alicerce           | 2       | 3          |
| 3                   | Viga baldrame           | 1       | 4          |
| 4                   | Alvenaria               | 4       | 5, 7, 8, 9 |
| 5                   | Piso                    | 1       | 6          |
| 6                   | Portas                  | 1       | 10         |
| 7                   | Janelas                 | 1       | 10         |
| 8                   | Telhado-Área de serviço | 1       | 10         |
| 9                   | Cobertura               | 2       | 10         |
| 10                  | Acabamentos             | 2       | 11         |
| 11                  | Limpeza                 | 1       | 12         |
| 12                  | Fim                     | 0       |            |

Fonte: o autor, 2019

FIGURA 26 – GRÁFICO ATIVIDADE-NO-NÓ DAS RELAÇÕES ENTRE AS ATIVIDADES PARA A EXECUÇÃO DE UMA UNIDADE HABITACIONAL



Fonte: o autor, 2019

Outros dados de entrada definidos para a execução do algoritmo foram a duração máxima  $D_T$  do projeto, igual a 30 meses, ou 660 dias (22 dias trabalhados por mês). O tempo de mobilização definido foi de 10 dias e restrição de recursos, ou seja, a quantidade máxima de recursos que poderia ser utilizada ao longo de toda a duração do projeto foi de 20 equipes. Como trata-se de um projeto simples, foi utilizada uma população menor, de 10 indivíduos, e diminuiu-se também o número máximo de gerações para 20. A execução do algoritmo foi muito rápida, durando cerca de 6 segundos.

Para a parte de programação das atividades, o algoritmo considera a duração para concluir cada atividade em todas as unidades habitacionais. Assim, se em uma das atividades a limitação de recursos for ultrapassada, todas as suas semelhantes serão reprogramadas, assim como as sucessoras. O número de equipes é determinado também pelo algoritmo, de acordo com os cálculos apresentados na sub-seção

O resultado final foi um arquivo no formato .csv com um total de 1010 atividades. A data inicial do projeto é definida pelo usuário e as outras são determinadas automaticamente. Nessa etapa final, para cada casa são desconsideradas as atividades “Início” e “Fim”, necessárias apenas para o funcionamento do Método de Programação Serial usado no cálculo na função de aptidão das soluções. Assim, cada unidade fica com 10 atividades, totalizando 1000

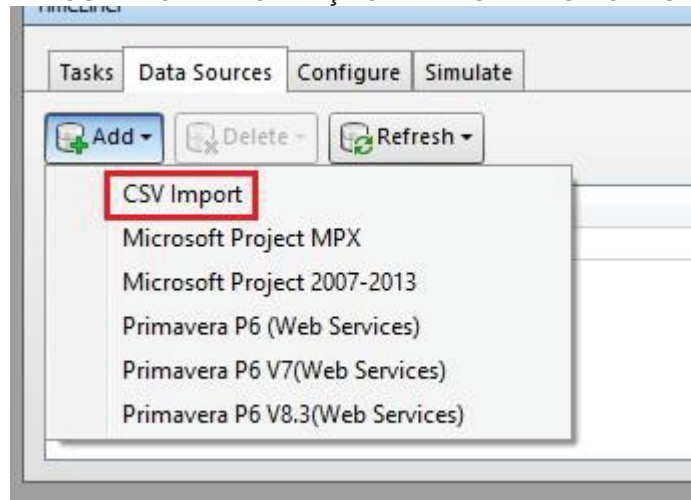
atividades. No entanto, no *Navisworks*, é necessário que todas as atividades do projeto estejam conectadas a alguma outra tarefa, para que elas apareçam na simulação. Assim, foi preciso modificar o programa mais uma vez para incluir 10 atividades finais sem duração, nas quais as atividades da última unidade foram ligadas.

FIGURA 27 – ARQUIVO .CSV GERADO PELO PROGRAMA

Fonte: o autor, 2019

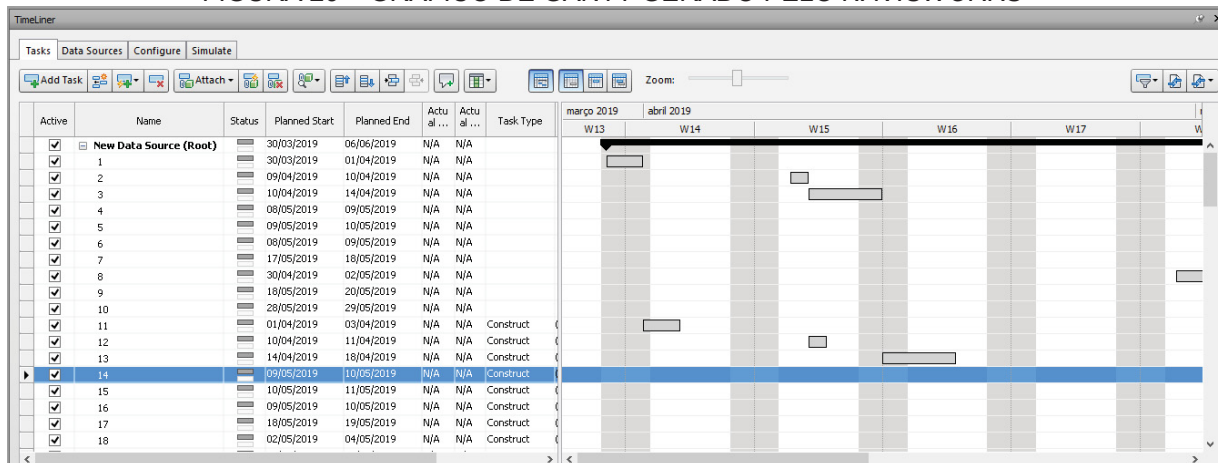


FIGURA 28 – IMPORTAÇÃO PARA O NAVISWORKS



Fonte: o autor, 2019

FIGURA 29 – GRÁFICO DE GANTT GERADO PELO NAVISWORKS

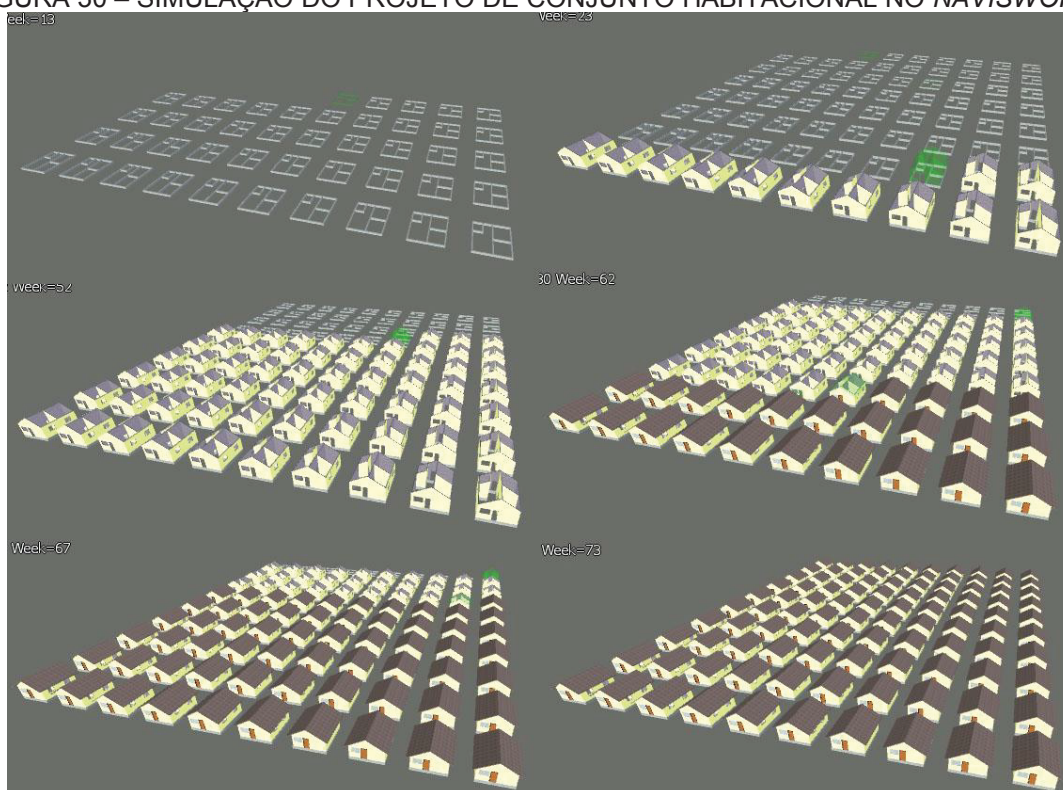


Fonte: o autor, 2019

Como é possível observar na Figura 29, a simulação pôde mostrar que o cronograma gerado respeita as relações de precedência entre as atividades e também demonstra que a lógica da linha de balanço foi bem implementada na programação, uma vez que a execução de atividades em uma unidade habitacional não depende da conclusão total das atividades nas unidades que começaram a ser construídas antes (Figura 28).



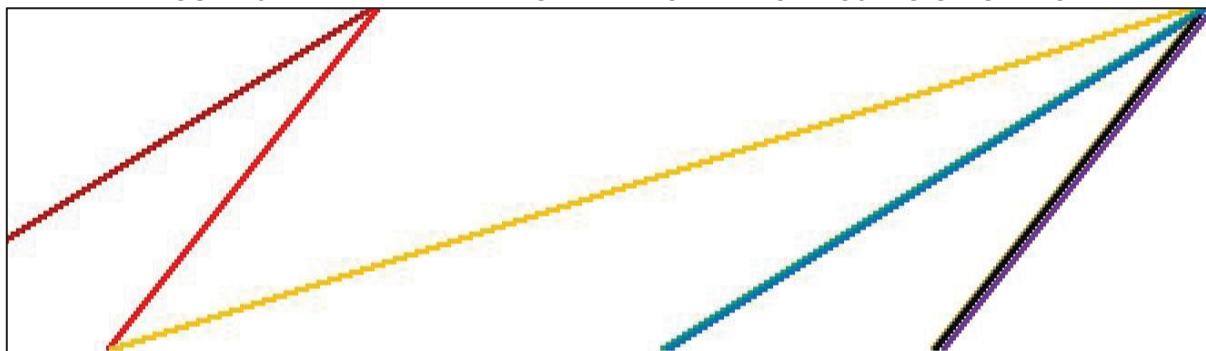
FIGURA 30 – SIMULAÇÃO DO PROJETO DE CONJUNTO HABITACIONAL NO NAVISWORKS



Fonte: o autor, 2019

O cronograma gerado pelo algoritmo também pode ser plotado, para observar melhor se os cálculos da Linha de Balanceamento foram bem implementados no processo de programação. A partir da análise do gráfico plotado (Figura 29), é possível concluir mais uma vez que a continuidade de trabalho na execução do projeto, característica crucial no método da Linha de Balanceamento, foi corretamente considerado na programação, uma vez que em todos os momentos há equipes sendo utilizadas.

FIGURA 31 – LINHA DE BALANCEAMENTO PARA O PROJETO SIMULADO



Fonte: o autor, 2019

#### 4.3 ANÁLISE DOS RESULTADOS

O método descrito neste trabalho contou com três etapas: a modelagem do problema, o módulo de otimização e o módulo de simulação. Todas estas fases foram realizadas e obtiveram-se os resultados esperados. O módulo de otimização contou com a implementação de duas variações do algoritmo genético, uma utilizando operadores clássicos descritos na literatura (GA) e outra fazendo uso de uma técnica de hibridização com outro tipo de meta-heurística, o *Simulated Annealing* (GASA).

Ao testar o desempenho dos dois algoritmos, o objetivo era investigar como outras variações do algoritmo genético influenciavam a busca de soluções, em relação à implementação clássica. No entanto, os resultados obtidos após numerosos experimentos computacionais mostraram que o algoritmo genético tradicional, por meio da estratégia de seleção por Elitismo, foi mais eficiente que o GASA tanto para encontrar soluções melhores, como quanto ao tempo gasto na realização deste processo. Na análise do comportamento da busca ao longo das gerações, foi observado que o GASA implementado, apesar de ser capaz de convergir rapidamente para boas soluções, oferece um grande risco de perdê-las, sobretudo em gerações iniciais quando a mecanismo de aceitação probabilística proveniente do *Simulated Annealing* ainda permite a passagem de mutações prejudiciais para a fase de recombinação.

Desse modo, é possível concluir para o módulo de otimização, que o operador de Elitismo é crucial para que o algoritmo tenda sempre à minimização das durações das instâncias de projeto. A capacidade de convergência do GASA, por sua vez, também pode ser muito vantajosa, ainda mais se for usada em conjunto com o Elitismo. Esta possibilidade não foi considerada neste trabalho, pois a intenção era testar dois tipos de técnicas diferentes na busca de melhoria das soluções. Além disso, o GASA requer praticamente o dobro do esforço computacional em relação ao GA, então seria necessário buscar alternativas que tornassem o cálculo da função de aptidão um pouco mais rápido, para que a utilização do GASA se mostrasse realmente oportuna.

Já no módulo de simulação, dois projetos foram modelado no *Autodesk Revit* e foram definidas também as atividades necessárias para a construção de cada um, as relações de precedência e as respectivas durações. Essas informações foram

usadas como dados de entrada para o algoritmo genético, que foi capaz de gerar uma solução aprimorada rapidamente.

Por meio da linguagem de programação *Python 2.7*, foi possível, então, gerar um arquivo no formato *.csv* com as datas de início e fim de todas as atividades dos projetos considerados. A possibilidade criar arquivos no formato *.csv* é muito vantajosa, pois *software* de gerenciamento de projetos como o *Autodesk Navisworks* são capazes de interpretá-los diretamente. Desse modo, com o sistema criado, em alguns segundos é possível obter um cronograma completo de todas as atividades a serem executadas, já considerando um número máximo de recursos. Dependendo da complexidade da rede de precedências, o algoritmo pode ser rodado diversas vezes, gerando diferentes soluções que podem ser simuladas, em busca de uma configuração que melhor atenda às prioridades dos executores.

A simulação realizada no *Autodesk Navisworks* ainda permitiu concluir que o cronograma gerado segue a lógica do método da Linha de Balanceamento, uma vez que, no modelo de simulação, as casas são construídas continuamente, sem a necessidade de que uma seja totalmente concluída para que as sucessoras comecem a ser executadas. Além disso, o sequenciamento de atividades em questão atende à restrição de recursos pré-determinada como parâmetro para o GA, fato constatado pelos próprios dados de utilização de recursos por dia fornecidos pelo algoritmo.

## 5. CONSIDERAÇÕES FINAIS

A presente dissertação procurou abordar o funcionamento e a relevância da aplicação de um procedimento meta-heurístico aliado à simulação BIM 4D no planejamento de obras de conjuntos habitacionais. A origem da pesquisa teve início nos trabalhos de Bezerra, Santos e Scheer (2018) e Bezerra e Scheer (2018), onde foi verificado o grande interesse de pesquisadores da área da construção civil em aplicar técnicas computacionais oriundas da Pesquisa Operacional em diversas fases do planejamento de obras, como a organização de atividades, alocação de recursos, *layout* do canteiro de obras e no controle de flutuações orçamentárias.

O foco em conjuntos habitacionais foi definido pela falta de estudos, tanto nacionais como internacionais, que apliquem essas técnicas a obras horizontais, que na grande maioria das vezes são a saída encontrada por governos de países em desenvolvimento para a promoção de políticas habitacionais. No Brasil, o programa “Minha Casa Minha Vida”, maior programa de habitação da América Latina, despertou grande interesse da academia desde sua criação, mas poucas pesquisas focam realmente em alternativas para aprimorar o processo de projeto e planejamento das obras, o que se mostra muitas vezes urgente, visto os problemas de restrições orçamentárias e de falta de qualidade das obras entregues, observados na literatura.

Hoje em dia, no Brasil, há ainda um aumento crescente do número de pessoas vivendo em condições precárias, sendo necessária a construção de aproximadamente 6,2 milhões de moradias para suprir o atual déficit habitacional. Sendo assim, alternativas que ajudem a tornar o planejamento e a execução destes empreendimentos mais rápida e racionalizada é muito relevante, não só para apenas construir mais casas, mas também para desenvolver o mercado de construção no país e conscientizar os profissionais envolvidos dos benefícios que a busca por novas técnicas inovadoras, sobretudo em fases pré-operacionais, podem trazer para o produto final entregue.

Sendo assim, para a definição do escopo da pesquisa, foi primeiramente realizada uma Revisão Sistemática da Literatura (Apêndice A), que buscou investigar quais técnicas computacionais oriundas da Pesquisa Operacional eram mais utilizadas no planejamento de obras de construção. Foram identificadas diversas técnicas, confirmando o quanto a agenda de pesquisa em questão é atual. No entanto,

observou-se o maior interesse acadêmico pelos procedimentos meta-heurísticos, sendo o Algoritmo Genético o mais utilizado.

Na revisão, também foram verificadas diversas publicações que tratavam da integração de *software* BIM com os procedimentos meta-heurísticos, sobretudo na utilização de dados provenientes de tais *software*, como também do seu uso como ferramenta de visualização e gerenciamento dos projetos. Sendo assim, decidiu-se também pela utilização de simulação BIM 4D como parte do sistema criado, permitindo integrar o aprimoramento do planejamento de obras a uma ferramenta moderna de visualização e controle de projetos.

A organização das atividades na obra no tempo, por sua vez, foi o tema mais abordado na aplicação dos procedimentos meta-heurísticos. A modelagem matemática para essa atividade, na grande maioria das publicações, foi inspirada no Problema de Programação de Projetos com Restrições de Recursos (*Resource-Constrained Project Scheduling Problem - RCPSP*), onde a ideia é buscar um sequenciamento de atividades com a menor duração possível, respeitando uma limitação de recursos, ou seja, um número máximo de recursos a ser utilizado durante toda a execução do projeto. O RCPSP se adequou bem à temática tratada neste trabalho, uma vez que projetos de habitação de interesse social lidam constantemente com escassez de recursos e precisam ser entregues o mais rápido possível.

Assim, o método para a condução da pesquisa foi desenvolvido em três etapas. Primeiramente, a modelagem do problema foi adaptada ao foco do trabalho, por meio da utilização do método da Linha de Balanceamento, ideal para o planejamento de longo prazo de projetos repetitivos, assim como são os conjuntos habitacionais. Para a implementação do algoritmo genético, foram definidas as formas de codificação das soluções, o procedimento de avaliação de aptidão, que se baseou em uma heurística de programação baseada em regra de prioridade, o Método Serial. Por fim, foram definidos os operadores genéticos de seleção, cruzamento e mutação, e os critérios de parada.

As abordagens de implementação de um algoritmo genético são muito amplas, pois esta técnica oferece grande possibilidade de modificação. Sendo assim, decidiu-se por testar a eficiência de duas estratégias híbridas para os operadores de seleção. A primeira, chamada de GA, utilizou os operadores de seleção por Elitismo e por Torneio. A segunda, o GASA, inspirada nos trabalhos de Bettemir (2009) e

Bettemir e Sonmez (2013), utilizou, além da seleção por torneio, um mecanismo de melhoria das soluções da população inicial por meio do conceito de *Simulated Annealing*, outra meta-heurística que permite a maior diversificação do espaço de busca e é capaz de melhorar as soluções ao longo das gerações. Apesar da estrutura dos algoritmos ser diferente, os operadores genéticos de mutação e cruzamento usados nos dois foram os mesmos, sendo estes o operador por Inversão e o operador de Um Ponto, respectivamente.

Após a implementação dos dois algoritmos, foram realizados numerosos experimentos com instâncias de projetos disponibilizados na base de dados para *benchmark* específica do RSPSP, a PSPLIB. Foram analisadas tanto as melhores e as piores soluções encontradas, assim como o tempo usado por cada algoritmo para encontrá-las. O GA se mostrou mais eficiente que o GASA tanto quanto à qualidade das soluções, como quanto ao tempo. No entanto, notou-se que o GASA tem uma tendência à convergência maior que o GA, tendo minimizado a duração das soluções em menos gerações em praticamente todas as instâncias utilizadas.

Na análise do comportamento dos algoritmos ao longo das gerações, foi concluído que a utilização da seleção por Elitismo é crucial para que boas soluções não sejam perdidas durante o processo de busca. Apesar do GASA ter a capacidade de melhorar as soluções, a falta da seleção por Elitismo faz com que muitas vezes o algoritmo descarte boas soluções em fases iniciais do processo de busca, quando a temperatura do sistema ainda não foi suficientemente reduzida, permitindo assim a aceitação de mutação piores.

Após a fase experimental, foi realizada a etapa aqui descrita como módulo de simulação. Nesta fase, a execução de um edifício e de conjunto habitacional de 100 casas foi simulada no *software Autodesk Navisworks*, com um cronograma gerado pelo GA. No caso do conjunto habitacional, como forma de demonstrar aplicação dos cálculos do método da Linha de Balanceamento no procedimento computacional implementado, foram definidas as relações de precedência para a execução de uma unidade habitacional, determinadas as durações das atividades e definidas uma restrição de recursos e uma duração máxima para o projeto. O algoritmo utilizou esses dados para gerar um cronograma que incluía todas as 1000 atividades necessárias para a conclusão da obra, na forma de um arquivo no formato .csv. O resultado da otimização pôde, então, ser lido pelo *Autodesk Navisworks*



diretamente, e as atividades foram conectadas ao modelo BIM previamente realizado, para que a simulação fosse desempenhada.

Na etapa de simulação, foi possível verificar que a programação gerada pelo algoritmo atende tanto às relações de precedência, quanto à lógica da Linha de Balanceamento. Além disso, foi possível traçar um histograma de utilização dos recursos para o projeto, onde foi possível observar que a restrição de recursos pré-determinada não foi violada e que a forma que o projeto foi planejado garante a manutenção da continuidade do trabalho durante toda a sua execução.

O sistema criado se mostra relevante principalmente no sentido de diminuir o esforço humano empregado no planejamento de obras de natureza repetitiva, uma vez que, para projetos simples como os de habitação popular, em poucos segundos é possível obter um cronograma para obra que considere tanto a continuidade do trabalho, quanto o atendimento a restrições de recursos. A simulação entra aqui como mais uma ferramenta para auxiliar na tomada de decisão, uma vez que várias possibilidades geradas pelo algoritmo podem ser testadas e mesmo modificadas por meio dos milhares de funcionalidades dos *software* de gerenciamento de obras baseados em BIM. A presente dissertação também busca chamar a atenção para a necessidade de maior utilização de *software* como esses, uma vez que são desenvolvidos especificamente para o mercado de construção e permitem visualização muito mais ampla da execução de uma obra, diferente de outros programas comerciais como *Microsoft Project*, *Primareva P6* ou mesmo o *Microsoft Excel*.

Desse modo, é possível considerar os objetivos do trabalho foram alcançados, uma vez que foi possível implementar duas variações do algoritmo genético e testar sua eficiência e ainda simular os cronogramas otimizados sem nenhuma dificuldade de compatibilização. O sistema ainda pode, no entanto, ser melhorado, uma vez que para projetos com grandes números de atividades o esforço computacional exigido se mostrou muito elevado. Além disso, outras questões além da duração ainda poderiam ser incluídas na implementação, que poderia servir de gatilho para o desenvolvimento de pesquisas futuras relacionadas a simulação, otimização e Pesquisa Operacional no setor de construção.

## 5.1 TRABALHOS FUTUROS

O tema abordado neste trabalho, não só no setor de construção, mas em várias outras indústrias, ainda é bastante atual e continuamente desenvolvido. Todos os dias, novos tipos de meta-heurísticas são criados e, para cada problema, uma ou outra pode gerar resultados melhores. A natureza combinatória dos problemas de programação de projetos torna muito difícil a determinação de soluções ótimas à medida que o número de atividades cresce, mas os procedimentos meta-heurísticos se mostram capazes de encontrar soluções satisfatórias, próximas do ótimo, na maioria dos casos.

O Algoritmo Genético já é implementado há muitas décadas para a resolução do RCPSP. No entanto, com o surgimento de novas meta-heurísticas, as possibilidades de hibridização são cada vez maiores, sendo esta uma questão importante a ser abordada em pesquisas futuras. Nos experimentos feitos com o algoritmo híbrido GASA, percebeu-se que a utilização da seleção por Elitismo poderia ter gerado melhores resultados do que os obtidos, por exemplo. No entanto, as possibilidades de modificação de um Algoritmo Genético são muitas, configurando um amplo campo de pesquisa.

A linguagem de programação utilizada neste trabalho, o *Python 2.7*, oferece diversas vantagens, sobretudo pela facilidade, em razão da sua sintática simples, de implementação dos algoritmos e na quantidade de pacotes e funções que já vem prontas para serem usadas em conjunto com a linguagem. Para a implementação feita neste trabalho, os elementos mais utilizados foram as listas e os dicionários, que, quando aplicados em conjunto com laços (*loops*) do tipo “*while*” (enquanto) ou “*for*” (para cada...), podem tornar o programa lento. Existem, no entanto, outros pacotes como o *Numpy* e o *Cython*, comumente utilizados para tornar implementações em *Python* mais velozes, que poderiam ser utilizados para tornar a execução dos algoritmos programados mais ágil, sem precisar utilizar outras linguagens com uma sintática mais complicada.

A modelagem do problema, por sua vez, poderia ser estendida no sentido de considerar não só atividades com o mesmo sentido de execução, a exemplo no que ocorre em edifícios com pavimentos-tipo, onde há atividades que são realizadas tanto no sentido térreo-cobertura como vice-versa. Além disso, outras questões além da duração, que já são largamente abordadas na literatura, poderiam ser incluídas à



modelagem, como a relação tempo-custo, a qualidade do projeto, sustentabilidade e planejamento de curto prazo.

Por fim, seria interessante também criar também interfaces do usuário que fossem mais amigáveis à utilização por pessoas leigas em programação, o que poderia abrir caminho para a criação de programas comerciais específicos para o aprimoramento do planejamento de empreendimentos de construção.

## REFERÊNCIAS

- ABBASIAN-HOSSEINI, S. A.; NIKAKHTAR, A.; GHODDOUSI, P. Verification of lean construction benefits through simulation modeling: A case study of bricklaying process. **KSCE Journal of Civil Engineering**, v. 18, n. 5, p. 1248–1260, 20 jun. 2014.
- AGDAS, D. et al. Utility of Genetic Algorithms for Solving Large-Scale Construction Time-Cost Trade-Off Problems. **Journal of Computing in Civil Engineering**, v. 32, n. 1, p. 04017072, jan. 2018.
- AGRAMA, Fatma A. Multi-objective genetic optimization of linear construction projects. **HBRC Journal**, v. 8, n. 2, p. 144-151, 2012.
- AL HATTAB, M.; ZANKOUL, E.; HAMZEH, F. R. Near-Real-Time Optimization of Overlapping Tower Crane Operations: A Model and Case Study. **Journal of Computing in Civil Engineering**, v. 31, n. 4, p. 05017001, jul. 2017.
- ALANJARI, P.; RAZAVIALAVI, S.; ABOURIZK, S. Hybrid Genetic Algorithm-Simulation Optimization Method for Proactively Planning Layout of Material Yard Laydown. **Journal of Construction Engineering and Management**, v. 141, n. 10, p. 06015001, out. 2015.
- ALTAF, M. S. et al. Integrated production planning and control system for a panelized home prefabrication facility using simulation and RFID. **Automation in Construction**, v. 85, p. 369–383, jan. 2018.
- AMIRI, R.; SARDROUD, J. M.; SOTO, B. G. DE. BIM-based Applications of Metaheuristic Algorithms to Support the Decision-making Process: Uses in the Planning of Construction Site Layout. **Procedia Engineering**, v. 196, p. 558–564, 2017.
- ARASHPOUR, M. et al. Off-site construction optimization: Sequencing multiple job classes with time constraints. **Automation in Construction**, v. 71, p. 262–270, nov. 2016.
- ARENALES, M; ARMENTANO, V; MORABITO, R; YANASSE, H. **Pesquisa operacional**. Rio de Janeiro: Elsevier, 2011.
- ARDITI, David; TOKDEMIR, Onur B.; SUH, Kangsuk. Challenges in line-of-balance scheduling. **Journal of construction engineering and management**, v. 128, n. 6, p. 545-556, 2002.
- BÄCK, Thomas; FOGEL, David B.; MICHALEWICZ, Zbigniew (Ed.). **Evolutionary computation 1: Basic algorithms and operators**. CRC press, 2018.
- BETTEMIR, Ö. H.; SONMEZ, R. Hybrid Genetic Algorithm with Simulated Annealing for Resource-Constrained Project Scheduling. **Journal of Management in Engineering**, v. 31, n. 5, p. 04014082, set. 2015.

BIOTTO, C. N.; FORMOSO, C. T.; ISATTO, E. L. Método para o uso da Modelagem BIM 4D na Gestão da Produção em Empreendimentos de Construção. In: SIMPÓSIO BRASILEIRO DE QUALIDADE DO PROJETO NO AMBIENTE CONSTRUÍDO, 3.; ENCONTRO DE TECNOLOGIA DE INFORMAÇÃO E COMUNICAÇÃO NA CONSTRUÇÃO, 6., 2013, Campinas. **Anais...** Porto Alegre: ANTAC, 2013.

BIRUK, S.; JAŚKOWSKI, P. Scheduling Linear Construction Projects with Constraints on Resource Availability. **Archives of Civil Engineering**, v. 63, n. 1, p. 3–15, 28 mar. 2017.

BRITO, D. M. de; FERREIRA, E. de A. M. Avaliação de estratégias para representação e análise do planejamento e controle de obras utilizando modelos BIM 4D. **Ambiente Construído**, Porto Alegre, v. 15, n. 4, p. 203-223, out./dez. 2015.

CAMPOS, R. B. A.; GUILHOTO, J. J. M. The socioeconomic impact of low-income housing programs: An interregional input-output model for the state of Sao Paulo and the rest of Brazil. **Habitat International**, v. 65, p. 59-69, 2017.

CHENG, M.-Y.; PRAYOGO, D.; TRAN, D.-H. Optimizing Multiple-Resources Leveling in Multiple Projects Using Discrete Symbiotic Organisms Search. **Journal of Computing in Civil Engineering**, v. 30, n. 3, p. 04015036, maio 2016.

CHO, C.-Y. et al. An optimal algorithm of the multi-lifting operating simulation for super-tall building construction. **Automation in Construction**, v. 35, p. 595–607, nov. 2013.

CHO, K.; HASTAK, M. Time and Cost–Optimized Decision Support Model for Fast-Track Projects. **Journal of Construction Engineering and Management**, v. 139, n. 1, p. 90–101, jan. 2013.

DONAGHY, M. M. **Civil society and participatory governance**: municipal councils and social housing programs in Brazil. New York: Taylor & Francis, 2013.  
EL-ABBASY, M. S.; ELAZOUNI, A.; ZAYED, T. Generic Scheduling Optimization Model for Multiple Construction Projects. **Journal of Computing in Civil Engineering**, v. 31, n. 4, p. 04017003, 1 jul. 2017.

ELNIMR, A.; FAGIAR, M.; MOHAMED, Y. Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changing site layout. **Automation in Construction**, v. 68, p. 235–248, ago. 2016.

EROL, H.; DIKMEN, I.; BIRGONUL, T. Measuring the impact of lean construction practices on project duration and variability: A simulation-based study on residential buildings. **JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT**, v. 23, n. 2, p. 241–251, 24 maio 2016.

FAGHIHI, V.; REINSCHMIDT, K. F.; KANG, J. H. Construction scheduling using Genetic Algorithm based on Building Information Model. **Expert Systems with Applications**, v. 41, n. 16, p. 7565–7578, nov. 2014.

- FUNG, I. W. H.; HUANG, C.; TAM, C. M. Application of GA optimization for solving precedent relationship problem in project scheduling. **Mathematical and Computer Modelling**, v. 57, n. 9–10, p. 2067–2081, maio 2013.
- GAN, X. et al. How affordable housing becomes more sustainable? A stakeholder study. **Journal of Cleaner Production**, v. 162, p. 427–437, 2017.
- GEEM, Z. W.; KIM, J. H.; LOGANATHAN, G. V.. A new heuristic optimization algorithm: harmony search. **simulation**, v. 76, n. 2, p. 60–68, 2001.
- GHODDOUSI, P. et al. Multi-mode resource-constrained discrete time–cost-resource optimization in project scheduling using non-dominated sorting genetic algorithm. **Automation in Construction**, v. 30, p. 216–227, mar. 2013.
- GIL, A. C. **Como elaborar projetos de pesquisa**. São Paulo: Editora Atlas, 2002.
- GIRAN, O.; TEMUR, R.; BEKDAŞ, G. Resource constrained project scheduling by harmony search algorithm. **KSCE Journal of Civil Engineering**, v. 21, n. 2, p. 479–487, 17 fev. 2017.
- GLEDSON, B. J; GREENWOOD, D. J. Surveying the extent and use of 4D BIM in the UK. **Journal of Information Technology in Construction (ITcon)**, Vol. 21, pg. 57–71, 2016. Disponível em: <http://www.itcon.org/2016/4>.
- GOLDBARG, M. C; GOLDBARG, E. G; LUNA, H. P. L. **Otimização combinatória e metaheurísticas**. Rio de Janeiro: Elsevier, 2016.
- GOLZARPOOR, H. et al. A non-queue-based paradigm in Discrete-Event-Simulation modelling for construction operations. **Simulation Modelling Practice and Theory**, v. 77, p. 49–67, set. 2017.
- GWAK, H.-S. et al. Exact Time–Cost Tradeoff Analysis in Concurrency-Based Scheduling. **Journal of Construction Engineering and Management**, v. 142, n. 10, p. 04016054, out. 2016.
- HAMMAD, A. W. A.; AKBARNEZHAD, A.; REY, D. A multi-objective mixed integer nonlinear programming model for construction site layout planning to minimise noise pollution and transport costs. **Automation in Construction**, v. 61, p. 73–85, jan. 2016.
- HILLIER, F. S; LIEBERMAN, G. J. **Introdução à pesquisa operacional**. Porto Alegre: AMGH, 2010.
- HU, W.; HE, X. An Innovative Time-Cost-Quality Tradeoff Modeling of Building Construction Project Based on Resource Allocation. **The Scientific World Journal**, v. 2014, p. 1–10, 2014.
- HUANG, Y. et al. Genetic algorithm–based method for the deadline problem in repetitive construction projects considering soft logic. **ascelibrary.org**, [s.d.].
- HYARI, Khalied; EL-RAYES, Khaled. Optimal planning and scheduling for repetitive construction projects. **Journal of Management in Engineering**, v. 22, n. 1, p. 11–19,

2006.

ICHIHARA, J. A. **Um método de solução heurístico para a programação de edifícios dotados de múltiplos pavimentos-tipo**. 188f. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1999.

JEENO, M; BRIJESH, P; DILEEPLAL, J; TINJUMOL, M. Multi objective optimization for scheduling repetitive projects using GA. In: GLOBAL COLLOQUIUM IN RECENT ADVANCEMENT AND EFFECTUAL RESEARCHES IN ENGINEERING, SCIENCE AND TECHNOLOGY, 2016, Kottayam, India. **Proceedings...** Kottayam: RAEREST 2016.

JIN, R. et al. Application of Case-Based Reasoning for Estimating Preliminary Duration of Building Projects. **Journal of Construction Engineering and Management**, v. 142, n. 2, p. 04015082, fev. 2016.

KHALILI, A.; CHUA, D. K. Integrated Prefabrication Configuration and Component Grouping for Resource Optimization of Precast Production. **Journal of Construction Engineering and Management**, v. 140, n. 2, p. 04013052, fev. 2014.

KHANZADI, M.; MOVAHEDIAN ATTAR, A.; BAGHERPOUR, M. Finding Optimum Resource Allocation to Optimizing Construction Project Time/Cost through Combination of Artificial Agents CPM and GA. **Periodica Polytechnica Civil Engineering**, v. 60, n. 2, p. 169–180, 2016.

KIM, J.-L.; ELLIS JR, R. D. Permutation-based elitist genetic algorithm for optimization of large-sized resource-constrained project scheduling. **Journal of construction engineering and management**, v. 134, n. 11, p. 904-913, 2008.

KOWALTOWSKI, D. C.C. K. et al. A critical analysis of research of a mass-housing programme. **Building Research & Information**, p. 1-18, 2018.

KOLISCH, R. Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. **European Journal of Operational Research**, v. 90, n. 2, p. 320-333, 1996.

KOLISCH, R; HARTMANN, S. Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis. In: **Project scheduling**. Springer, Boston, MA, 1999. p. 147-178.

KOLISCH, R; SPRECHER, A. PSPLIB – a project scheduling problem library. **European Journal of Operational Research**, v. 96, p. 205-216, jan. 1996.

LIMMER, C. V. **Planejamento, orçamentação e controle de projetos e obras**. Rio de Janeiro: LTC, 1997.

LIU, H.; AL-HUSSEIN, M.; LU, M. BIM-based integrated approach for detailed construction scheduling under resource constraints. **Automation in Construction**, v. 53, p. 29–43, maio 2015.

LU, W.; OLOFSSON, T. Building information modeling and discrete event simulation: Towards an integrated framework. **Automation in Construction**, v. 44, p. 73–83, ago. 2014.

MARZOUK, M. et al. An optimization algorithm for simulation-based planning of low-income housing projects. **Journal of Advanced Research**, v. 1, n. 4, p. 291-300, 2010.

MARZOUK, M. et al. A simulation optimisation tool for planning of low-income housing projects. **Civil Engineering and Environmental Systems**, v. 31, n. 1, p. 51–63, 2 jan. 2014.

MENDES JÚNIOR, R. **Programação da produção na construção de edifícios de múltiplos pavimentos**. 2f. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1999.

MENESI, W.; HEGAZY, T. Multimode Resource-Constrained Scheduling and Leveling for Practical-Size Projects. **Journal of Management in Engineering**, v. 31, n. 6, p. 04014092, nov. 2015.

MOON, H. et al. BIM-Based Construction Scheduling Method Using Optimization Theory for Reducing Activity Overlaps. **Journal of Computing in Civil Engineering**, v. 29, n. 3, p. 04014048, maio 2015.

MOSELHI, Osama; EL-RAYES, Khaled. Scheduling of repetitive projects with cost optimization. **Journal of Construction Engineering and Management**, v. 119, n. 4, p. 681-697, 1993.

OLIVA, C. A.; GRANJA, A. D. An Investigation Into Collaborative Practices in Social Housing Projects as a Precondition for Target Value Design Adoption. In: **Formoso, CT & Tzortzopoulos, P., 21th Annual Conference of the International Group for Lean Construction. Fortaleza, Brazil**. 2013. p. 31-2.

PONZ-TIENDA, J. L. et al. Improved Adaptive Harmony Search algorithm for the Resource Leveling Problem with minimal lags. **Automation in Construction**, v. 77, p. 82–92, maio 2017.

PONZ-TIENDA, J. L.; SALCEDO-BERNAL, A.; PELLICER, E. A Parallel Branch and Bound Algorithm for the Resource Leveling Problem with Minimal Lags. **Computer-Aided Civil and Infrastructure Engineering**, v. 32, n. 6, p. 474–498, jun. 2017.  
POTGIETER, I. J.; VAN ROOYEN, G. C. Maximizing Slack in Resource-Constrained Schedules: A Heuristic Approach. **Journal of Computing in Civil Engineering**, v. 30, n. 6, p. 04016018, nov. 2016.

PRAYOGO, D. et al. Optimization model for construction project resource leveling using a novel modified symbiotic organisms search. **Asian Journal of Civil Engineering**, v. 19, n. 5, p. 625–638, 24 jul. 2018.

REDA, R. M. RPM: Repetitive project modeling. **Journal of Construction Engineering and Management**, v. 116, n. 2, p. 316-330, 1990.

ROSENDO, M. Um algoritmo de otimização por nuvem de partículas para resolução de problemas combinatórios. 93f. Dissertação (Mestrado em Informática) – Programa de Pós-Graduação em Informática, Universidade Federal do Paraná, Curitiba, 2010.

RUBIN, G. R.; BOLFE, S. A. The development of social housing in Brazil. **Ciência e Natura**, v. 36, n. 2, p. 201-213, 2014.

SOMERS, K.; BAUD, I. My House, My Life: Decision-making processes and local citizen participation in housing project Minha Casa, Minha Vida in Salvador da Bahia. N-AERUS XIV. 2015.

TAGHADDOS, H. et al. Simulation-Based Multiagent Approach for Scheduling Modular Construction. **Journal of Computing in Civil Engineering**, v. 28, n. 2, p. 263–274, mar. 2014.

TURRIONI, J. B.; MELLO, C. H. P. **Metodologia de pesquisa em engenharia de produção**. Itajubá: UNIFEI, 2012.

VAN LAARHOVEN, Peter JM; AARTS, Emile HL. Simulated annealing. In: **Simulated annealing: Theory and applications**. Springer, Dordrecht, 1987. p. 7-15.

VON ALLENRIEDER, A. **Pesquisa operacional**. Rio de Janeiro: Almeida Neves – Editores Ltda, 1971.

WALL, M. B. **A genetic algorithm for resource-constrained scheduling**. 62f. Tese (Doutorado em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Massachusetts Institute of Technology, Cambridge, 1996.

WANG, J. et al. A BIM-based approach for automated tower crane layout planning. **Automation in Construction**, v. 59, p. 168–178, nov. 2015.

WANG, Z; HU, H; GONG, J. Framework for modeling operational uncertainty to optimize offsite production scheduling of precast components. **Automation in Construction**, v. 86, p. 69-80, 2018.

YANG, X. Nature-Inspired Metaheuristic Algorithms Second Edition. 2010.

ZHANG, L.; DU, J.; ZHANG, S. Solution to the Time-Cost-Quality Trade-off Problem in Construction Projects Based on Immune Genetic Particle Swarm Optimization. **Journal of Management in Engineering**, v. 30, n. 2, p. 163–172, mar. 2014.



## APÊNDICE A

### REVISÃO SISTEMÁTICA DA LITERATURA

Neste item encontra-se descrito o processo de Revisão Sistemática da Literatura, apresentando-se uma análise sobre as ferramentas utilizadas nas pesquisas e suas áreas de aplicação. Em seguida, é apresentada uma breve descrição de cada trabalho, a fim de entender os exatos processos onde as técnicas foram utilizadas.

#### 1. MÉTODO

Dresch, Lacerda e Antunes Júnior (2015) definem a Revisão Sistemática da Literatura como “uma etapa fundamental da condução de pesquisas científicas”. Trata-se de estudos secundários utilizados para mapear, avaliar criticamente e agregar os resultados de estudos primários relevantes acerca de uma questão de pesquisa específica, bem como identificar lacunas a serem preenchidas, resultando em um relatório coerente ou em uma síntese. A expressão *sistemática* significa que a revisão deve seguir um método explícito e planejado, de modo a garantir que o trabalho seja isento de viés. Com o uso deste método, é esperado que a consolidação e agregação dos estudos primários resultem em novo conhecimento (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015).

Ainda de acordo com Dresch, Lacerda e Antunes Júnior (2015), o primeiro passo na realização de qualquer revisão sistemática é a definição do tema central, para que então possa ser formulada a questão principal que a pesquisa irá responder, servindo como ponto de partida para entender o trabalho e seu contexto.

Desse modo, o tema principal definido para a presente revisão sistemática é a utilização de técnicas computacionais, comumente aplicadas na área de Pesquisa Operacional, como forma de buscar o aprimoramento do planejamento de obras. O trabalho procurará responder, então, o seguinte questionamento: “Quais técnicas computacionais estão sendo utilizadas de modo a auxiliar o processo de tomada de decisão na indústria de construção? ”.

Procedeu-se então com a definição dos termos de busca a serem utilizados na revisão. Com base em algumas publicações de referência (HILLIER; LIEBERMAN, 2010; KÖNIG et al., 2012; HU; MOHAMED, 2014; BAKHT; EL-DIRABY, 2015; FAGHIHI et al., 2015), foram escolhidos termos que se relacionassem com pesquisa



operacional, metaheurísticas, programação, aprimoramento de processos e tomada de decisão. Optou-se por pesquisar no idioma inglês, de modo a alcançar um universo maior de resultados.

A pesquisa foi realizada em três bases de dados eletrônicas: Science Direct, ASCE Library e Scopus. Em cada uma das bases foram feitas oito pesquisas utilizando os conjuntos de termos de busca anteriormente definidos. Restringiu-se a busca apenas a artigos científicos publicados entre 2013 e 2018, que tivessem os termos adotados no título, resumo e palavras chave. Foi possível, assim, encontrar mais de 14300 publicações (Tabela 1).

Tabela 1 – Resultados na busca inicial

| TERMOS DE BUSCA                            | SCIENCE DIRECT | SCOPUS | ASCE LIBRARY |
|--|----------------|--------|--------------|
| OPERATIONS RESEARCH CONSTRUCTION INDUSTRY  | 128            | 323    | 1996         |
| OPERATIONAL RESEARCH CONSTRUCTION INDUSTRY | 43             | 130    | 773          |
| METAHEURISTICS CONSTRUCTION INDUSTRY       | 5              | 8      | 20           |
| PROGRAMMING CONSTRUCTION INDUSTRY          | 207            | 160    | 1997         |
| ALGORITHM CONSTRUCTION INDUSTRY            | 134            | 405    | 839          |
| OPTIMIZATION CONSTRUCTION INDUSTRY         | 272            | 451    | 1226         |
| SIMULATION CONSTRUCTION INDUSTRY           | 198            | 589    | 1640         |
| DECISION MAKING CONSTRUCTION INDUSTRY      | 192            | 617    | 1951         |
| TOTAL                                      |                | 14304  |              |

Fonte: o autor, 2018

Foi realizada, então, a filtragem dos artigos encontrados, de modo a encontrar aqueles que melhor se encaixavam à proposta do trabalho. Primeiramente, foram eliminados todos os artigos duplicados e, em seguida, suprimidos os trabalhos cujo título não se relacionava com o questionamento da pesquisa. A terceira filtragem foi feita, então, com base na leitura dos *abstracts* de cada uma das publicações restantes. Com este resultado, procedeu-se com a análise do conteúdo dos artigos, tendo sido possível eliminar mais alguns que não se adequavam ao trabalho. O resultado final encontrado foi de 38 artigos (Tabela 2).

Tabela 2 – Filtragens das publicações

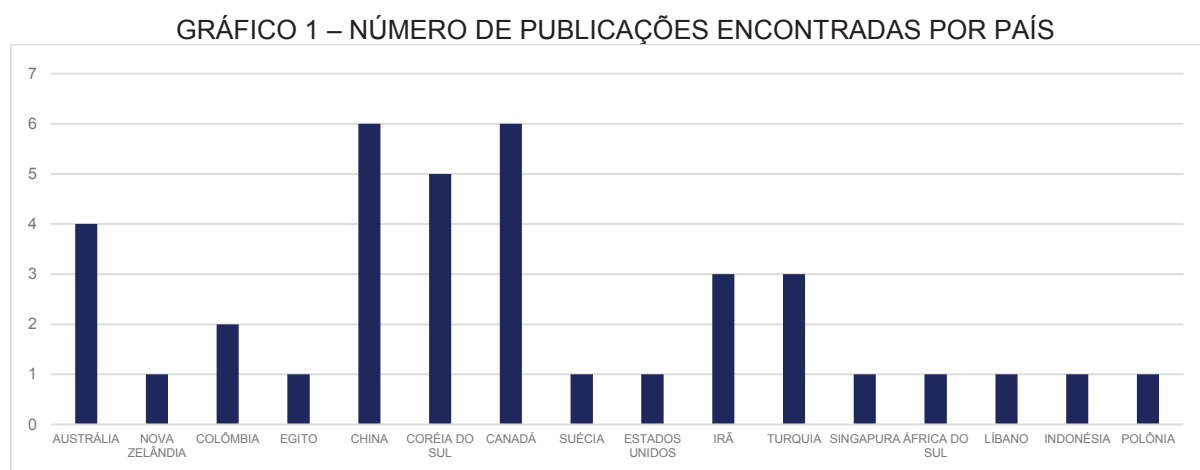
|                                   |      |
|-----------------------------------|------|
| <b>1ª FILTRAGEM - DUPLICADOS</b>  | 5808 |
| <b>2ª FILTRAGEM - TÍTULO</b>      | 325  |
| <b>3ª FILTRAGEM - ABSTRACT</b>    | 70   |
| <b>FILTRAGEM FINAL - CONTEÚDO</b> | 38   |

Fonte: o autor, 2018

Com os resultados encontrados, foi realizada uma breve análise bibliométrica, que consiste em uma técnica quantitativa e estatística de medição de

produção e disseminação do conhecimento científico (ARAÚJO, 2006). Desse modo, foi observada a origem das publicações encontradas e seu ano de publicação, a fim de avaliar em quais países o tema em questão tem sido mais pesquisado e se este pode ser considerado um assunto atual.

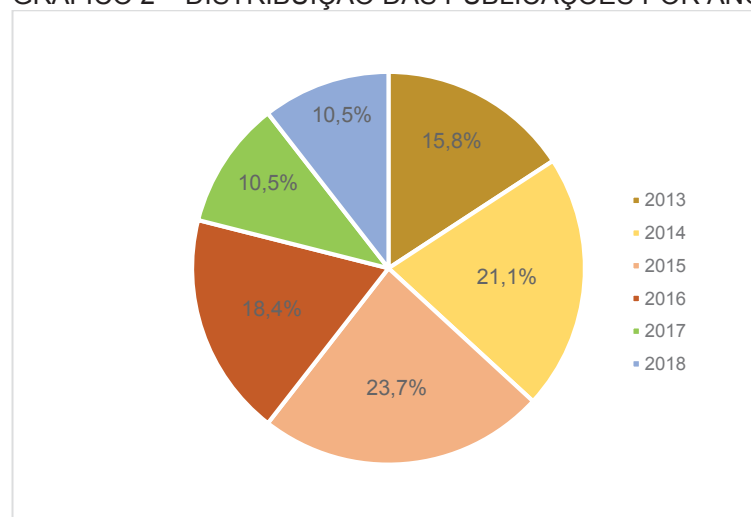
Foi possível observar que as publicações encontradas são provenientes de diversos países, com representantes de todos os continentes. Destacam-se, no entanto, com maior número de publicações, a China e o Canadá. Coreia do Sul, Austrália, Irã e Turquia também apresentaram um número relevante de publicações, chamando a atenção para a existência de pesquisadores dedicados à área de pesquisa contemplada na revisão nestes países. Ainda foram encontradas publicações oriundas da América do Sul, Europa e África (Gráfico 1).



Fonte: o autor, 2018

No período em que a pesquisa foi restrita, foi possível observar um maior número de publicações nos anos de 2015 e 2016. No entanto, também foram encontradas pesquisas relevantes publicadas em 2018, sendo possível afirmar que o tema em questão ainda constitui uma área de pesquisa atual (Gráfico 2).

GRÁFICO 2 – DISTRIBUIÇÃO DAS PUBLICAÇÕES POR ANO



Fonte: o autor, 2018

## 2. RESULTADOS E DISCUSSÃO

A partir da análise das publicações selecionadas, foi possível identificar diversas abordagens para permitir uma melhor tomada de decisão em empreendimentos de construção. Foi possível dividir as técnicas utilizadas pelos pesquisadores em seis grupos: metaheurísticas, simulação computacional, programação linear, programação não-linear, modelos de otimização de redes e outras abordagens menos específicas (Quadro 1).

Quadro 1 – Técnicas utilizadas nas pesquisas selecionadas (continua)

| AUTOR  | META-HEURÍSTICAS | SIMULAÇÃO | PROG. LINEAR | PROG. NÃO-LINEAR | OTIM. REDES | OUTROS |
|--|------------------|-----------|--------------|------------------|-------------|--------|
| Wang et al. (2015)                           | •                |           |              |                  |             |        |
| Hammad, Akbarnezhad e Rey (2015)             |                  |           |              | •                |             |        |
| Golzarpour et al. (2017)                     |                  | •         |              |                  |             |        |
| Polz-Tienda, Salcedo-Bernal, Pellicer (2016) |                  |           | •            |                  |             |        |
| Marzouk et al. (2014)                        | •                | •         |              |                  |             |        |
| Hu e He (2014)                               | •                |           |              | •                |             |        |
| Cho et al. (2013)                            |                  |           | •            |                  |             |        |

Fonte: o autor, 2018

Quadro 1 – Técnicas utilizadas nas pesquisas selecionadas (continuação)

| AUTOR                                   | META-<br>HEURÍSTICAS | SIMULAÇÃO | PROG.<br>LINEAR | PROG. NÃO-<br>LINEAR | OTIM.<br>REDES | OUTROS |
|---|----------------------|-----------|-----------------|----------------------|----------------|--------|
| Jin et al. (2015)                       |                      |           |                 |                      |                | •      |
| Fung, Huang e Tam (2013)                | •                    |           |                 |                      |                |        |
| Moon et al. (2015)                      | •                    |           |                 |                      |                | •      |
| Liu; Al-Hussein; Lu (2015)              | •                    | •         |                 |                      |                |        |
| Lu; Olofsson (2014)                     |                      | •         |                 |                      |                |        |
| Faghihi; Reinschmidt; Kang (2014)       | •                    |           |                 |                      |                |        |
| Gwak et al. (2016)                      |                      |           |                 |                      | •              |        |
| Khanzadi, Attar e Begherpour (2015)     | •                    |           |                 |                      |                |        |
| Huang, Hu e Gong (2018)                 | •                    | •         |                 |                      |                |        |
| El-Abbasy, Elazouni e Zayed (2017)      | •                    |           |                 |                      |                |        |
| Huang, Zou e Zhang (2016)               | •                    |           | •               |                      |                |        |
| Bettemir e Sonmez (2013)                | •                    |           |                 |                      |                |        |
| Alanjari, RazaviAlavi e AbouRizk (2015) | •                    | •         |                 |                      |                |        |
| Polz-Tienda et al. (2016)               | •                    |           |                 |                      |                |        |
| Khalili e Chua (2013)                   |                      |           | •               |                      |                |        |
| Altaf et al. (2018)                     | •                    | •         |                 |                      |                |        |
| Potgieter e van Rooyen (2016)           |                      |           |                 |                      | •              |        |
| Erol, Dikmen e Birgonul (2015)          |                      | •         |                 |                      |                |        |
| Ghoddousi et al. (2013)                 | •                    |           |                 |                      |                |        |
| Menezi e Hegazy (2014)                  |                      |           |                 |                      | •              |        |
| Hattab, Zankoul e Hamzeh (2017)         | •                    | •         |                 |                      |                |        |

Fonte: o autor, 2018

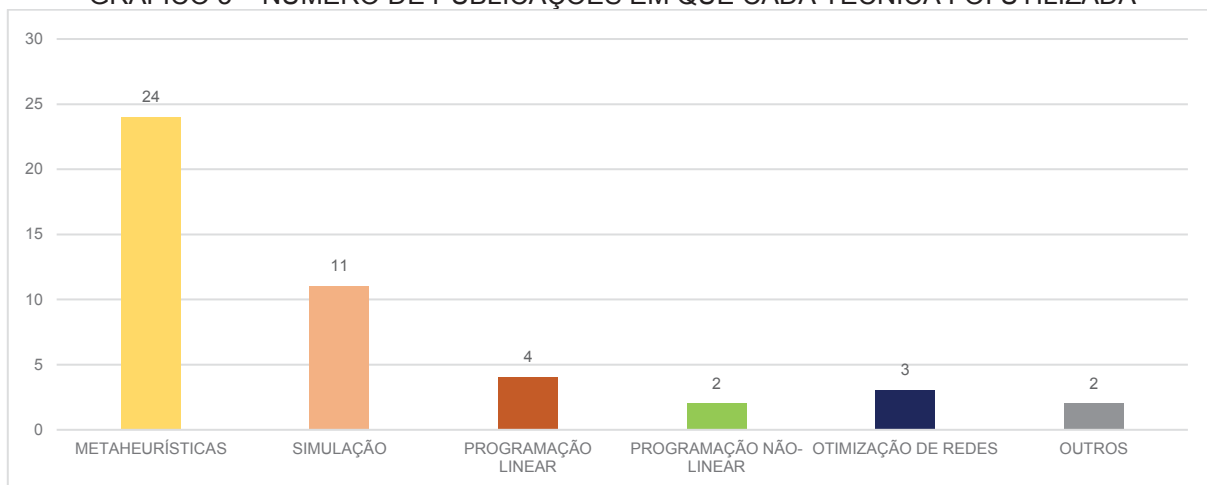
Quadro 1 – Técnicas utilizadas nas pesquisas selecionadas (conclusão)

| AUTOR   | META-<br>HEURÍSTICAS | SIMULAÇÃO | PROG.<br>LINEAR | PROG. NÃO-<br>LINEAR | OTIM.<br>REDES | OUTROS |
|---|----------------------|-----------|-----------------|----------------------|----------------|--------|
| Hattab, Zankoul e Hamzeh (2017)                 | •                    | •         |                 |                      |                |        |
| Arashpour et al. (2016)                         | •                    |           |                 |                      |                |        |
| Prayogo et al. (2018)                           | •                    |           |                 |                      |                |        |
| Cheng, Prayogo e Tran (2015)                    | •                    |           |                 |                      |                |        |
| Giran, Temur e Bekdas (2016)                    | •                    |           |                 |                      |                |        |
| Biruk e Jakowski (2017)                         |                      |           | •               |                      |                |        |
| Taghaddos et al. (2014)                         |                      | •         |                 |                      |                |        |
| Zhang, Du e Zhang (2014)                        | •                    |           |                 |                      |                |        |
| Cho e Hastak (2013)                             | •                    |           |                 |                      |                |        |
| Agdas et al. (2018)                             | •                    |           |                 |                      |                |        |
| Abbasian-Hosseini, Nikakhtar e Ghoddousi (2014) |                      | •         |                 |                      |                |        |

Fonte: o autor, 2018

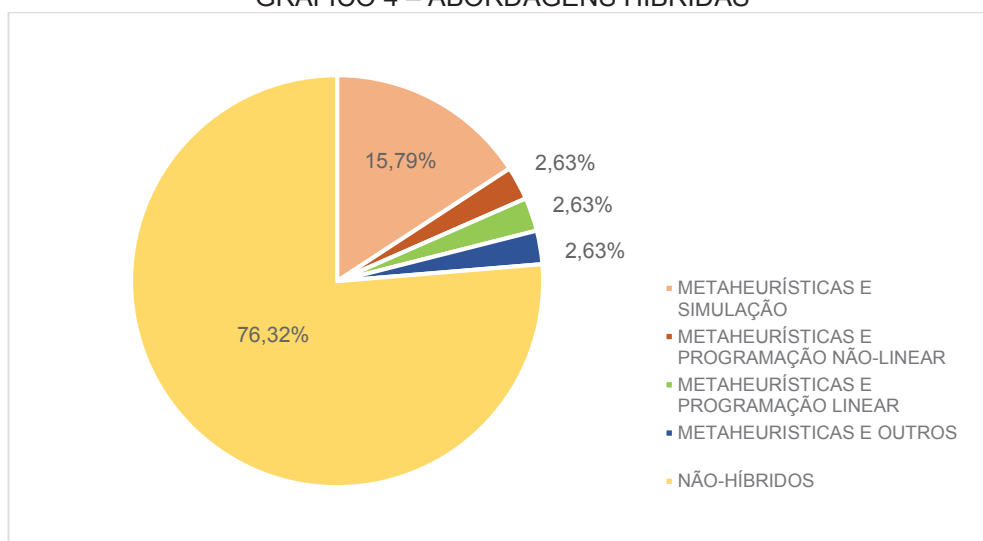
Por meio do levantamento realizado no Quadro 1, foi possível observar uma predominância pela utilização de algoritmos metaheurísticos para subsidiar a tomada de decisões. Também foi identificado um grande número de publicações que abordavam a utilização de softwares de simulação para modelar os sistemas estudados. Alguns estudos implementaram também sistemas híbridos, onde mais de uma técnica era utilizada. Observou-se que em todas essas abordagens híbridas, os pesquisadores utilizaram metaheurísticas para melhorar o desempenho da técnica principal utilizada no estudo (Gráfico 3 e 4).

GRÁFICO 3 – NÚMERO DE PUBLICAÇÕES EM QUE CADA TÉCNICA FOI UTILIZADA



Fonte: o autor, 2018

GRÁFICO 4 – ABORDAGENS HÍBRIDAS



Fonte: o autor, 2018

Foi realizada ainda mais uma análise das publicações, desta vez de acordo com o tipo de processo que se buscou aprimorar em cada um dos estudos. Foi possível então identificar, desta vez, quatro grupos principais: utilização de equipamentos, configuração (*layout*) das instalações do canteiro de obras, planejamento de atividades e gerenciamento de recursos (Quadro 2).

Quadro 2 – Tipos de processos aprimorados pelas técnicas utilizadas (continua)

| AUTOR                            | UTILIZAÇÃO DE EQUIPAMENTOS | LAYOUT DO CANTEIRO | PLANEJAMENTO DE ATIVIDADES | GERENCIAMENTO DE RECURSOS |
|----------------------------------|----------------------------|--------------------|----------------------------|---------------------------|
| Wang et al. (2015)               | •                          |                    |                            |                           |
| Hammad, Akbarnezhad e Rey (2015) |                            | •                  |                            |                           |

Fonte: o autor, 2018

Quadro 2 – Tipos de processos aprimorados pelas técnicas utilizadas (continuação)

| AUTOR  | UTILIZAÇÃO DE EQUIPAMENTOS | LAYOUT DO CANTEIRO | PLANEJAMENTO DE ATIVIDADES | GERENCIAMENTO DE RECURSOS |
|--|----------------------------|--------------------|----------------------------|---------------------------|
| Golzarpour et al. (2017)                     |                            |                    | •                          |                           |
| Polz-Tienda, Salcedo-Bernal, Pellicer (2016) |                            |                    |                            | •                         |
| Marzouk et al. (2014)                        |                            |                    | •                          | •                         |
| Hu e He (2014)                               |                            |                    |                            | •                         |
| Cho et al. (2013)                            | •                          |                    |                            |                           |
| Jin et al. (2015)                            |                            |                    | •                          |                           |
| Fung, Huang e Tam (2013)                     |                            |                    | •                          |                           |
| Moon et al. (2015)                           |                            |                    | •                          |                           |
| Liu; Al-Hussein; Lu (2015)                   |                            |                    | •                          | •                         |
| Lu; Olofsson (2014)                          |                            |                    | •                          |                           |
| Faghihi; Reinschmidt; Kang (2014)            |                            |                    | •                          |                           |
| Gwak et al. (2016)                           |                            |                    | •                          |                           |
| Khanzadi, Attar e Begherpour (2015)          |                            |                    |                            | •                         |
| Huang, Hu e Gong (2018)                      |                            |                    | •                          |                           |
| El-Abbasy, Elazouni e Zayed (2017)           |                            |                    | •                          | •                         |
| Huang, Zou e Zhang (2016)                    |                            |                    | •                          |                           |
| Bettemir e Sonmez (2013)                     |                            |                    | •                          | •                         |
| Alanjari, RazaviAlavi e AbouRizk (2015)      |                            | •                  |                            |                           |
| Polz-Tienda et al. (2016)                    |                            |                    |                            | •                         |
| Khalili e Chua (2013)                        |                            |                    |                            | •                         |

Fonte: o autor, 2018

Quadro 2 – Tipos de processos aprimorados pelas técnicas utilizadas (conclusão)

| AUTOR   | UTILIZAÇÃO DE EQUIPAMENTOS | LAYOUT DO CANTEIRO | PLANEJAMENTO DE ATIVIDADES | GERENCIAMENTO DE RECURSOS |
|---|----------------------------|--------------------|----------------------------|---------------------------|
| Altaf et al. (2018)                             |                            |                    | •                          |                           |
| Potgieter e van Rooyen (2016)                   |                            |                    | •                          | •                         |
| Erol, Dikmen e Birgonul (2015)                  |                            |                    | •                          |                           |
| Ghoddousi et al. (2013)                         |                            |                    | •                          | •                         |
| Menezi e Hegazy (2014)                          |                            |                    | •                          | •                         |
| Hattab, Zankoul e Hamzeh (2017)                 | •                          |                    |                            |                           |
| Arashpour et al. (2016)                         |                            |                    | •                          |                           |
| Prayogo et al. (2018)                           |                            |                    |                            | •                         |
| Cheng, Prayogo e Tran (2015)                    |                            |                    |                            | •                         |
| Giran, Temur e Bekdas (2016)                    |                            |                    | •                          | •                         |
| Biruk e Jakowski (2017)                         |                            |                    | •                          | •                         |
| Taghaddos et al. (2014)                         |                            |                    | •                          | •                         |
| Zhang, Du e Zhang (2014)                        |                            |                    | •                          |                           |
| Cho e Hastak (2013)                             |                            |                    | •                          |                           |
| Agdas et al. (2018)                             |                            |                    | •                          |                           |
| Abbasian-Hosseini, Nikakhtar e Ghoddousi (2014) |                            |                    | •                          |                           |

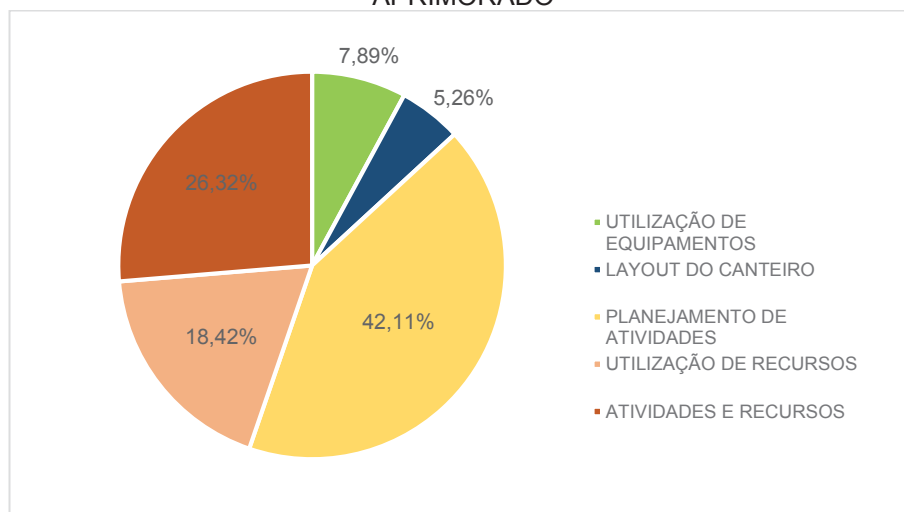
Fonte: o autor, 2018

Foi possível observar que a grande maioria das publicações buscou encontrar uma configuração melhor para o sequenciamento das atividades de projeto, de modo a diminuir sua duração, custos e, muitas vezes, aumentar sua qualidade. Há um foco grande também no aprimoramento da utilização de recursos, o que em muitos



dos artigos é feito em conjunto com o processo de planejamento de atividades. Foi possível perceber que há uma forte relação entre esses dois processos nas pesquisas desta área (Gráfico 5).

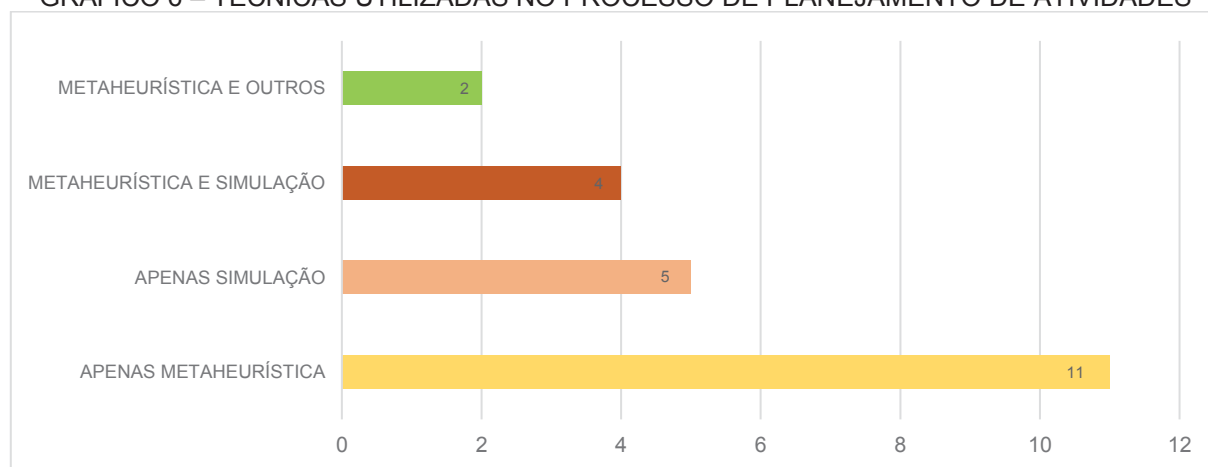
GRÁFICO 5 – DISTRIBUIÇÃO DAS PUBLICAÇÕES DE ACORDO COM O PROCESSO APRIMORADO



Fonte: o autor, 2018

Assim, observou-se que, na área de planejamento de obras, o uso destas técnicas computacionais pode ser aplicado principalmente no sequenciamento de atividades, procurando aperfeiçoar cronogramas de obra e a relação tempo-custo dos projetos. Verificou-se também que, como na maioria das publicações selecionadas, a técnica mais utilizada para esse intuito são as metaheurísticas, seguidas da simulação computacional. Além disso, foi possível observar que a maioria das abordagens híbridas implementadas no processo em questão utilizaram *software* de simulação e algoritmos metaheurísticos em conjunto (Gráfico 6).

GRÁFICO 6 – TÉCNICAS UTILIZADAS NO PROCESSO DE PLANEJAMENTO DE ATIVIDADES



Fonte: o autor, 2018

Desse modo, os artigos foram ainda analisados de acordo com a divisão feita no Gráfico 5, onde se procurou entender como as técnicas utilizadas em cada estudo funcionam, o objeto de estudo que foi aperfeiçoado pela sua utilização e as vantagens que a implementação de cada técnica pode trazer para o mercado de construção. Cada ferramenta será, também, conceituada, pois constituem elementos específicos das áreas de tecnologia da informação e pesquisa operacional, com os quais profissionais de outras áreas podem não estar familiarizados.

## 2.1 UTILIZAÇÃO DE EQUIPAMENTOS

Na análise dos artigos selecionados na revisão, foi observado que alguns estudos não se preocupavam exatamente com o planejamento da obra no sentido de definir conjuntos de atividades e alocar recursos, mas com a tomada de decisões sobre a melhor forma de utilizar as máquinas e equipamentos que subsidiam o processo construtivo.

Com esse intuito, Wang et al. (2015) usam um tipo de metaheurística chamada *Firefly Algorithm* ou algoritmo de vaga-lume. O termo *heurístico* significa, de maneira geral, “encontrar” ou “descobrir por tentativa e erro”. Assim, a utilização de algoritmos heurísticos é capaz de encontrar soluções de qualidade para problemas de otimização em um período de tempo razoável. No entanto, esses tipos de algoritmos não garantem a obtenção de soluções ótimas o tempo todo. São, no entanto, boas ferramentas para quando não se busca necessariamente as melhores soluções, mas sim aquelas que são satisfatórias e facilmente alcançáveis (YANG, 2010).

Por sua vez, algoritmos metaheurísticos são recursos mais desenvolvidos. O prefixo *meta* remete a “além” ou “em um nível mais alto”, tendo essas ferramentas um melhor desempenho geral que algoritmos heurísticos simples. Isto se deve ao fato de que os algoritmos metaheurísticos incorporam um grau de aleatoriedade à busca por soluções, fazendo com que essas ferramentas sejam adequadas para problemas de otimização global (YANG, 2010).

O desenvolvimento de algoritmos metaheurísticos é muitas vezes inspirado por fenômenos da natureza. O *Firefly Algorithm* por sua vez, imita o comportamento de vagalumes que são atraídos uns pelos outros de acordo com a intensidade da luz que emitem (WANG et al., 2015).

Wang et al. (2015) utilizam tal técnica em conjunto com modelos BIM para encontrar e simular configurações aprimoradas para as posições das guias e das estações de abastecimento dentro do canteiro de obras. Desse modo, os autores buscam balancear a carga de trabalho das guias e diminuir seu tempo de operação, assim como minimizar conflitos espaciais entre as guias e outros equipamentos móveis no canteiro.

É criado, assim, um sistema que usa informações espaciais e quantitativas provenientes do modelo BIM como dados de entrada, com os quais o algoritmo citado buscará soluções aprimoradas para o problema. Essas soluções são então simuladas em um *software* BIM 4D, para que os processos possam ser melhor analisados. O sistema é considerado inovador pelo fato de utilizar informações criadas automaticamente pelo *software* BIM, eliminando a necessidade de entrada manual de dados (WANG et al., 2015).

Por sua vez, Cho et al. (2013) tratam de melhorias na utilização de elevadores para transportar operários e materiais entre os andares de edifícios altos. Os autores implementam um algoritmo para calcular as melhores rotas para os elevadores, em situações em que materiais e trabalhadores precisem ser transportados entre os andares da obra ao mesmo tempo.

Para isso, Cho et al. (2013) modelam o processo como um problema de programação linear inteira. Esta técnica refere-se à formulação de modelos matemáticos para descrever um problema real. O adjetivo *linear* significa que todas as funções matemáticas do modelo são *funções lineares*, enquanto o termo programação pode ser interpretado como um sinônimo de *planejamento*. Assim, a programação linear consiste no planejamento de atividades para obter um resultado ótimo, ou seja, um resultado que atinja o melhor objetivo especificado (de acordo com o modelo matemático) (HILLIER; LIEBERMAN, 2010). Por sua vez, a programação linear inteira se refere à classe de problemas de otimização em que algumas ou todas as variáveis de decisão são inteiras.

Por fim, Al Hattab, Zankoul e Hamzeh (2017) também tratam da operação de guias, dessa vez usando a técnica de planejamento *Look-ahead* aliada à simulação de eventos discretos e metaheurística. Segundo Martinez (1996), simulação é um processo de modelagem que imita sistemas dinâmicos, reais ou imaginários. Consiste

na criação de um modelo de um sistema, no qual serão executados experimentos a fim de avaliar o possível comportamento do processo em uma situação real.

O conceito de simulação de eventos discretos (*Discrete Event Simulation*) é a base do *software* utilizado em Al Hattab, Zankoul e Hamzeh (2017) e também de outras plataformas utilizadas nas pesquisas da presente revisão. Neste tipo de simulação, supõe-se que o estado de um sistema muda instantaneamente em tempos específicos, caracterizados por eventos. Assim, a maioria dos processos construtivos pode ser modelado de maneira eficaz utilizando esta abordagem (MARTINEZ, 1996).

Assim, os autores desenvolvem um sistema que primeiramente implementa um planejamento *Look-ahead*, dividindo a programação geral da obra em atividades menores, para que se possa analisar a necessidade de utilização de guias em cada atividade e seu tempo de uso. Em seguida, o *software Anylogic7* é usado para modelar o sistema e simulá-lo, de modo a identificar gargalos e sobreposições de atividades. Junto ao *software*, os autores ainda utilizam um *plug-in* chamado *OptQuest*, que é capaz de aprimorar ainda mais os resultados da simulação por meio da aplicação de vários algoritmos metaheurísticos.

## 2.2 LAYOUT DO CANTEIRO DE OBRAS

Outra abordagem verificada em alguns dos artigos foi o aprimoramento das posições das instalações do canteiro de obras, de modo a permitir a melhor movimentação de recursos humanos, materiais e equipamentos dentro de um espaço limitado. De fato, as publicações na área partem do princípio que o espaço dentro do canteiro de obras é um recurso limitado, e portanto tem que ser bem projetado para evitar prejuízos à produtividade (AMIRI; SARDROUD; DE SOTO, 2017).

Hammad, Akbarnezhad e Rey (2016) usam como técnica a programação não-linear. Esta técnica tem um conceito basicamente igual ao de programação linear, porém aqui a modelagem matemática conta com funções *não-lineares*. O problema a ser solucionado no estudo é diminuição da poluição sonora causada pelo movimento de equipamentos pesados no canteiro. A função não-linear presente no modelo é o cálculo dos níveis sonoros emitidos pelas operações, que possuem uma escala logarítmica. O modelo matemático formulado pelos autores leva em consideração dois objetivos: a diminuição do custo de transporte de recursos entre as instalações e a diminuição da poluição sonora percebida na vizinha do canteiro.

No estudo de Alanjari, RazaviAlavi e AbouRizk (2015), por sua vez, busca-se tomar decisões quanto às melhores regiões do canteiro de obras para posicionar materiais recebidos e também quanto ao cronograma de entregas destes insumos ao empreendimento. Os autores desenvolvem um sistema híbrido, onde combinam simulação computacional a um algoritmo genético, outro tipo de metaheurística largamente utilizada (AMIRI; SARDROUD; DE SOTO, 2017). Os algoritmos genéticos são modelos ou abstrações da evolução biológica baseados na teoria da seleção natural de Charles Darwin, usando conceitos como cruzamento, mutação e seleção no estudo de sistemas adaptativos e artificiais. Esses “operadores genéticos” formam uma parte essencial do algoritmo genético como uma estratégia de solução de problemas (YANG, 2010).

Assim, o sistema criado em Alanjari, RazaviAlavi e AbouRizk (2015) é capaz de aprimorar o posicionamento de materiais baseado em seu consumo, modelar a remoção de materiais do canteiro da forma mais próxima possível à realidade e integrar o cronograma de entregas ao sistema de otimização, considerando assim o dinamismo do fluxo de fornecimento. Como resultado, foi possível prever as quantidades certas de materiais para cada etapa da obra, facilitando o planejamento do canteiro para a recepção.

## 2.3 GERENCIAMENTO DE RECURSOS

Um dos maiores focos de pesquisa verificados na revisão foi o de gerenciamento de recursos, onde os pesquisadores utilizam técnicas computacionais de modo a definir a melhor forma para utilizar os recursos durante a duração de um projeto. Busca-se o melhor sequenciamento de atividades para que não haja grandes flutuações no uso de recursos, dentro de um período de finalização pré-determinado.

Ponz-Tienda, Salcedo-Bernal & Pellicer (2016) tratam do chamado Problema de Alocação de Recursos (*Resource Leveling Problem* – RLP), onde o objetivo é justamente encontrar uma sequência de atividades que maximize a eficiência do consumo de recursos, minimizando a variabilidade (flutuações), respeitando a duração limite do empreendimento. Como técnica, os autores modelaram o problema por meio de programação linear, e implementaram o algoritmo em um projeto de edifício residencial.

Em Polz-Tienda et al. (2017), o mesmo problema abordado em Ponz-Tienda, Salcedo-Bernal & Pellicer (2016) é tratado. Desta vez, no entanto, é usada como técnica a metaheurística *Harmony Search* ou Busca Harmônica. Trata-se de uma metaheurística baseada em harmonia musical, ou seja, na ideia de que a música tocada por um conjunto de instrumentos vai se tornando mais harmônica de acordo com a prática (GEEM; KIM; LOGANATHAN, 2001). São realizados testes de validade no algoritmo, que é posteriormente implementado em um projeto de um edifício de 15 pavimentos.

Cheng, Prayogo e Tran (2015) e Prayogo et al. (2018) também tratam do RLP, utilizando como técnica a metaheurística *Symbiotic Organisms Search*, inspirada pelas interações de dependência vistas entre organismos na natureza, conhecidas como simbiose. Em Cheng, Prayogo e Tran (2015) é abordada a alocação de múltiplos recursos a vários projetos realizados ao mesmo tempo, enquanto em Prayogo et al. (2018) o foco é na melhor utilização de recursos humanos.

Em Hu e He (2014), o problema a ser solucionado é a relação conflitante entre tempo, custo e qualidade de projeto. No estudo, procura-se minimizar a duração e o custos do projeto e, ao mesmo tempo, aumentar sua qualidade. Para isso, os autores se baseiam na utilização de recursos durante o empreendimento, partindo do pressuposto de que cada recurso sofre variações de qualidade, custo e produtividade, e que estes são fatores determinantes para o tempo de finalização da construção. Assim, é utilizado mais uma vez um algoritmo genético para buscar por soluções aprimoradas para o problema, que é aplicado a um projeto de uma casa de três pavimentos.

Khanzadi, Movahedian e Begherpour (2015) também abordam a alocação de recursos no projeto, dessa vez utilizando a técnica de Modelagem Baseada em Agentes (*Agent-Based Modeling – ABM*) em conjunto com o método CPM (*Critical Path Method*) para definir as inter-relações entre as atividades levando em consideração a interação entre cada atividade, os recursos necessários para realizá-la e as condições do ambiente de projeto (sob o aspecto espacial e gerencial).

A Modelagem Baseada em Agentes parte do princípio de que o comportamento dos agentes envolvidos no projeto é afetado pelas suas características e interações com outros agentes, aprendendo com suas experiências e se adaptando ao ambiente em que estão inseridos. Por meio da modelagem

individual dos agentes, os efeitos das variações de comportamento podem ser traçados. Se isso é alcançado, o comportamento do sistema como um todo pode ser determinado. Com essa técnica, por meio da decomposição dos sistemas, agente por agente e interação por interação, a precisão do modelo é aprimorada (KHANZADI; MOVAHEDIAN; BEGHERPOUR, 2015).

Desse modo, os autores combinam o método CPM com a ABM de modo que a rede de atividades considere a interação entre as atividades, recursos necessários e ambiente de execução. Com isso, o objetivo final dos autores é a busca por uma alocação de recursos aprimorada para o projeto por meio de um algoritmo genético. O sistema é implementado em um pequeno projeto composto por oito atividades (KHANZADI; MOVAHEDIAN; BEGHERPOUR, 2015).

Khalili e Chua (2013) utilizam programação linear para gerenciar a produção de elementos construtivos pré-fabricados de concreto. Os autores desenvolvem um modelo de programação linear para o problema de planejamento operacional e utilização de formas na linha de produção, quando um alto grau de pré-fabricação como componentes complexos é adotado. O modelo desenvolve o cronograma de produção de modo a atender às demandas do canteiro de obras, considerando a complexidade dos elementos pré-fabricados, planejamento da utilização de formas e restrições da fábrica (fornecimento de materiais, área de produção, etc).

## 2.4 PLANEJAMENTO DE ATIVIDADES

Na RSL realizada, as técnicas de aprimoramento de processos e tomada de decisão foram aplicadas com maior frequência ao planejamento de atividades, isto é, à definição da melhor sequência de tarefas para as obras, simulação de processos e busca pela redução da duração e dos custos de projeto.

Golzarpour et al. (2017) utilizam como técnica a simulação de eventos discretos, de modo a melhorar a tomada de decisão quanto ao sequenciamento de atividades. Os autores desenvolvem uma aplicação própria, adaptada à forma como as operações em empreendimentos de construção se realizam. Segundo os autores, os *software* de simulação convencionais não são capazes de modelar de maneira eficiente os processos de construtivos, pois foram desenvolvidos para a indústria de manufatura. Assim, é desenvolvida uma aplicação baseada no conceito de



*Hierarchical Control Conceptual Modeling*, que é capaz de modelar melhor as sequências de atividades de construção e utilização de recursos.

Por sua vez, Jin et al. (2015) utilizam como técnica o Raciocínio Baseado em Casos (*Case-Based Reasoning* – CBR). Trata-se de uma técnica de resolução de problemas que se baseia no conhecimento obtido em situações passadas, e reutiliza essas informações para resolver novos problemas. A técnica é realizada em quatro etapas: recuperar o caso mais similar possível, reutilizar a informação recuperada para solucionar um novo problema, revisar a nova solução de modo a identificar diferenças em relação ao caso que fora recuperado e armazenar a nova solução para solucionar problemas futuros. A técnica é aplicada na determinação, em fases pré-operacionais, da duração da execução do empreendimento, e é validada em projetos de conjuntos habitacionais.

Em Fung e Tam (2013), procura-se aprimorar o cronograma de atividades por meio de um algoritmo genético. Para isso, os autores usam como dados de entrada planilhas eletrônicas com dados sobre quantidades de materiais e custos unitários, além de informações sobre as precedências entre as tarefas. Segundo os autores, a combinação do algoritmo genético com as planilhas eletrônicas permite a melhor visualização do processo de modelagem, fortalecendo a confiança e o fácil entendimento dos processos executados.

Moon et al. (2015) utilizam um sistema de planejamento de obras baseado em BIM, capaz de detectar e reorganizar atividades sobrepostas na rede CPM (caminho crítico). O *software* desenvolvido integra o modelo 3D com o planejamento de atividades. O processo desenvolvido ocorre em três etapas: identificação das atividades que coincidem e do seu nível de sobreposição, análise dos riscos desta sobreposição para o desempenho da obra por meio de lógica *fuzzy* e aplicação de um algoritmo genético para aprimorar a programação da obra.

O sistema cria um novo planejamento para a obra com sobreposição minimizada de atividades sem que a duração total do projeto seja alterada. Como variável para o algoritmo genético, foi usado o número de dias em que as atividades poderiam ser alteradas sem que o tempo de execução aumentasse. É possível ainda analisar a nova configuração por meio de simulação 4D no *software* criado (MOON et al., 2015).



Faghihi, Reinschmidt e Kang (2014) também propõem um meio de gerar o planejamento da construção automaticamente a partir de informações extraídas de um modelo BIM. As informações sobre as interdependências entre os elementos do modelo são organizadas em formato de matriz, que pode ser lida por um algoritmo genético desenvolvido, garantindo a construtibilidade do projeto. No entanto, a complexidade do modelo BIM limita o processo desenvolvido, pois quanto mais elementos existirem no projeto, mais complicado será gerar uma programação correta para a obra.

Lu e Olofsson (2014), por sua vez, utilizam o *software* de simulação *Simio* para avaliar os melhores cenários para a execução de componentes da construção. São extraídos quantitativos do modelo BIM, que dão suporte para a simulação das atividades. O sistema facilita a tomada de decisões, uma vez que permite que os profissionais possam avaliar rapidamente os melhores métodos construtivos. É possível analisar, ainda, as implicações de mudanças no projeto de maneira automatizada, sem necessidade de atualizar o modelo de simulação manualmente.

Gwak et al. (2016) procuram diminuir o tempo e, conseqüentemente, os custos do empreendimento, por meio da execução concomitante das atividades críticas do projeto. Para isso, é criado um método computacional que identifica uma solução global, ou seja, o conjunto de atividades que podem ser sobrepostas, de modo que não seja necessário alocar recursos adicionais. Assim, profissionais de planejamento podem tomar decisões, antes e durante a execução, sobre quais atividades podem ser realizadas ao mesmo tempo, diminuindo assim a duração do projeto.

O planejamento da linha de produção de elementos construtivos pré-fabricados é abordado por Wang, Hu e Gong (2018), onde busca-se aprimorar os processos por meio de um sistema híbrido, composto por simulação computacional e um algoritmo genético. Os cronogramas de produção e entregas são usados como dados de entrada para rodar o algoritmo genético, que a cada iteração gera programações distintas. Cada uma dessas possibilidades é, então, modelada em um *software* de simulação, permitindo avaliação de cada configuração e a escolha de uma solução ótima.

O planejamento de elementos pré-fabricados também é tratado em Altaf et al. (2018), onde é desenvolvido um sistema que utiliza identificação por

radiofrequência (RFID), simulação computacional e algoritmos metaheurísticos. O RFID é utilizado para obter informações em tempo real sobre a linha de produção. Esses dados dão suporte, então, ao modelo de simulação computacional, que é aprimorado pelo uso de metaheurísticas.

São usados no trabalho dois tipos de algoritmo: otimização por núvem de partículas ou *Particle Swarm Optimization* e *Simulated Annealing*. O primeiro trata-se de um sistema multiagente para resolver problemas de otimização complexos. Uma característica marcante deste algoritmo é a forma como os agentes, ou partículas, interagem entre si. Eles apresentam um comportamento colaborativo, simulando um compartilhamento de experiências e cooperando entre si em busca das melhores soluções (ROSENDO, 2010). Por sua vez, o algoritmo *Simulated Annealing*, é inspirado no processo de resfriamento de metais, que, quando feito por meio de decréscimos lentos de temperatura, dá origem a cristais com uma estrutura molecular bem organizada (LUKE, 2015).

Assim, o sistema criado permite obter informações sem tempo real para o aprimoramento da automático do planejamento da produção de painéis pré-fabricados. É exposto ainda que o algoritmo *Particle Swarm Optimization* tem um melhor desempenho que *Simulated Annealing* (ALTAF et al., 2018).

Arashpour et al. (2016) também utilizam metaheurísticas para aprimorar o sequenciamento de atividades na linha de produção de elementos pré-fabricados, porém não especificam qual algoritmo foi implementado.

Em Huang, Zou e Zhang (2016), é abordada a possibilidade de sequenciar atividades sob diferentes configurações. Segundo os autores, existem conjuntos de atividades de podem ser realizados em diferentes ordens ou mesmo simultaneamente. Porém, para cada sequência, o custo de transferência de recursos de uma atividade para a outra muda, dependendo das distâncias entre os locais de execução ou da disponibilidade de equipes. Assim, os autores implementam um algoritmo genético para tentar achar a melhor solução possível para o problema, objetivando a diminuição dos custos de projeto.

Por sua vez, Erol, Dikmen e Birgonul (2015) investigam os impactos de práticas de construção enxuta nos empreendimentos e os comparam com projetos executados de maneira convencional, por meio do Método de Monte-Carlo, técnica de simulação probabilística. Os autores concluem que a aplicação de princípios enxutos

tem grande potencial para diminuição da duração dos projetos, principalmente em atividades como a construção de paredes.

Em um estudo parecido, Abbasian-Hosseini, Nikakhtar e Ghoddousi (2014) também estudam o impacto dos princípios da construção enxuta no planejamento de atividades, abordando também a construção de paredes de alvenaria. Os autores no entanto, utilizam o *software* de simulação de eventos discretos *ARENA* para modelar e analisar tal processo.

Zhang, Du e Zhang (2014) buscam um aprimoramento multi-objetivo, onde busca-se diminuir a duração e os custos de projeto, ao passo que sua qualidade é maximizada. Para isso, os autores utilizam uma combinação dos algoritmos genético e de otimização por nuvem de partículas. A modelagem matemática do problema leva em consideração a relação entre a qualidade do projeto e o tempo de execução de cada atividade. O sistema é validado em um projeto de edifício comercial de três pavimentos.

Em Agdas et al. (2017), chama-se a atenção para a eficiência no uso de metaheurísticas para aprimorar a relação conflitante tempo-custo em projetos reais de construção. Os autores implementam, então, um algoritmo genético e fazem análises quanto ao custo computacional, tempo para encontrar a solução e requisitos de *software* e *hardware* necessários para encontrar soluções ótimas em projetos com grande número de atividades.

O método de planejamento *Fast-track*, que consiste em planejar a fase de projeto ocorrendo simultaneamente à execução, é abordado por Cho e Hastak (2013) como uma maneira de diminuir a duração e os custos de um empreendimento. Assim, os autores criam modelos matemáticos para calcular a duração e os custos totais de da construção de acordo com o método *Fast-track*, que são utilizados na função-objetivo de um algoritmo genético. O sistema melhora a tomada de decisão sobre adotar ou não o método *Fast-track* para o planejamento de projetos de construção.

## 2.5 ATIVIDADES E RECURSOS

Ao analisar as publicações selecionadas na revisão, foi constatado que uma grande parte não tratava apenas do planejamento de atividades, mas procurava fazê-lo em conjunto com a busca por melhorias na eficiência da utilização de recursos.

Marzouk et al. (2014) desenvolvem um estudo que busca melhorar a utilização de recursos escassos por agências governamentais na construção de habitações de interesse social. Para isso, os autores usam o *software* de simulação *Stroboscope* para modelar o processo específico de construção deste tipo de projeto e buscam aprimorar o tempo e o custo total de execução por meio de um algoritmo genético.

Em El-Abbasy, Elazouni e Zayed (2017), utiliza-se também um algoritmo genético, como forma de auxiliar na distribuição de recursos e organizar a execução de múltiplos empreendimentos ocorrendo simultaneamente. Para isso, os autores fazem uma modelagem matemática multi-objetivo, buscando aprimorar a duração, custos totais, custos financeiros, lucro, flutuações de utilização de recursos e picos de demanda. O estudo é considerado inovador por considerar tanto aspectos financeiros quanto relativos à utilização de recursos em um mesmo modelo.

Por sua vez, Liu, Al-Hussein e Lu (2015) desenvolvem um sistema onde o recurso central utilizado é o software de simulação *Symphony.NET*, que é ligado a uma ferramenta de análise de interdependências entre elementos e ao algoritmo de otimização por nuvem de partículas por meio da API (*Application Programming Interface*) da plataforma BIM *Autodesk Revit*. O algoritmo é usado pelos autores para encontrar a melhor sequência de atividades, baseado na duração total do projeto extraída do modelo de simulação.

A busca de soluções para o Problema de Programação de Projetos com Restrições de Recursos (*Resource-Constrained Scheduling Problem – RCPSP*) é abordado por Bettemir e Sonmez (2014) por meio da aplicação das metaheurísticas *Simulated Annealing* e algoritmo genético. O objetivo geral do RCPSP é planejar a execução de projetos com recursos limitados por meio da determinação das datas de início de cada atividade, de modo que tanto as restrições de precedência quanto de recursos sejam satisfeitas e que a duração total do projeto seja minimizada. Segundo os autores, o RCPSP é crucial para o planejamento de projetos de construção, uma vez que restrições de recursos têm grande influência sobre esse tipo de empreendimento (BETTEMIR; SONMEZ, 2014).

Desse modo, os autores desenvolvem um algoritmo híbrido a partir das duas metaheurísticas supracitadas e, por meio de testes, provam que a abordagem adotada é mais eficiente do que a utilização de *softwares* de planejamento disponíveis

no mercado (BETTEMIR; SONMEZ, 2014). Em um estudo mais recente, Giran, Temur e Bekdas (2016) também obtêm resultados satisfatórios para o problema, mas desta vez utilizando a metaheurística *Harmony Search* ou Busca Harmônica. Já em Biruk e Jakowski (2017), soluções para o problema são obtidas por meio de programação linear.

Em Potgieter e van Rooyen (2016), trata-se da criação de um modelo computacional que faça a alocação de recursos em um projeto de modo a aumentar a folga livre. Segundo os autores, a folga total de um projeto e sua distribuição entre as atividades é influenciada diretamente pela alocação de recursos. Assim, o modelo desenvolvido tem como função-objetivo a maximização da folga total do projeto, de modo criar uma ferramenta que beneficie a execução de projetos com períodos curtos de duração e permitir que interrupções imprevistas em atividades não prejudiquem a duração total do empreendimento.

Por sua vez, Menesi e Hegazy (2014) tratam do chamado Problema de Programação de Projetos com Restrições de Recursos e Múltiplos Modos de Execução (*Multi-Mode Resource-Constrained Project Scheduling Problem – MRCPSP*), que busca o mesmo objetivo que RCPSP, mas considera que as atividades podem ser executadas de modos diferentes, com requisitos distintos de recursos. Os autores desenvolvem um modelo no software de otimização *ILOG CPLEX*, com o qual obtêm bons resultados para projetos com mais de 2000 atividades, em um período curto de tempo.

O MRCPSP também é abordado em Ghoddousi et al. (2013), onde além da duração e da utilização de recursos, os autores ainda buscam aprimorar os custos de projeto, por meio da implementação de um algoritmo genético. Por fim, Taghaddos et al. (2014) estudam a aplicação do MRCPSP a fabricação de construções modulares, que são unidades habitacionais feitas inteiramente na fábrica e transportadas ao canteiro para instalação. Para isso, os autores criam um *software* de simulação próprio, que usa como base informações definidas em processos licitatórios.

## APÊNDICE B

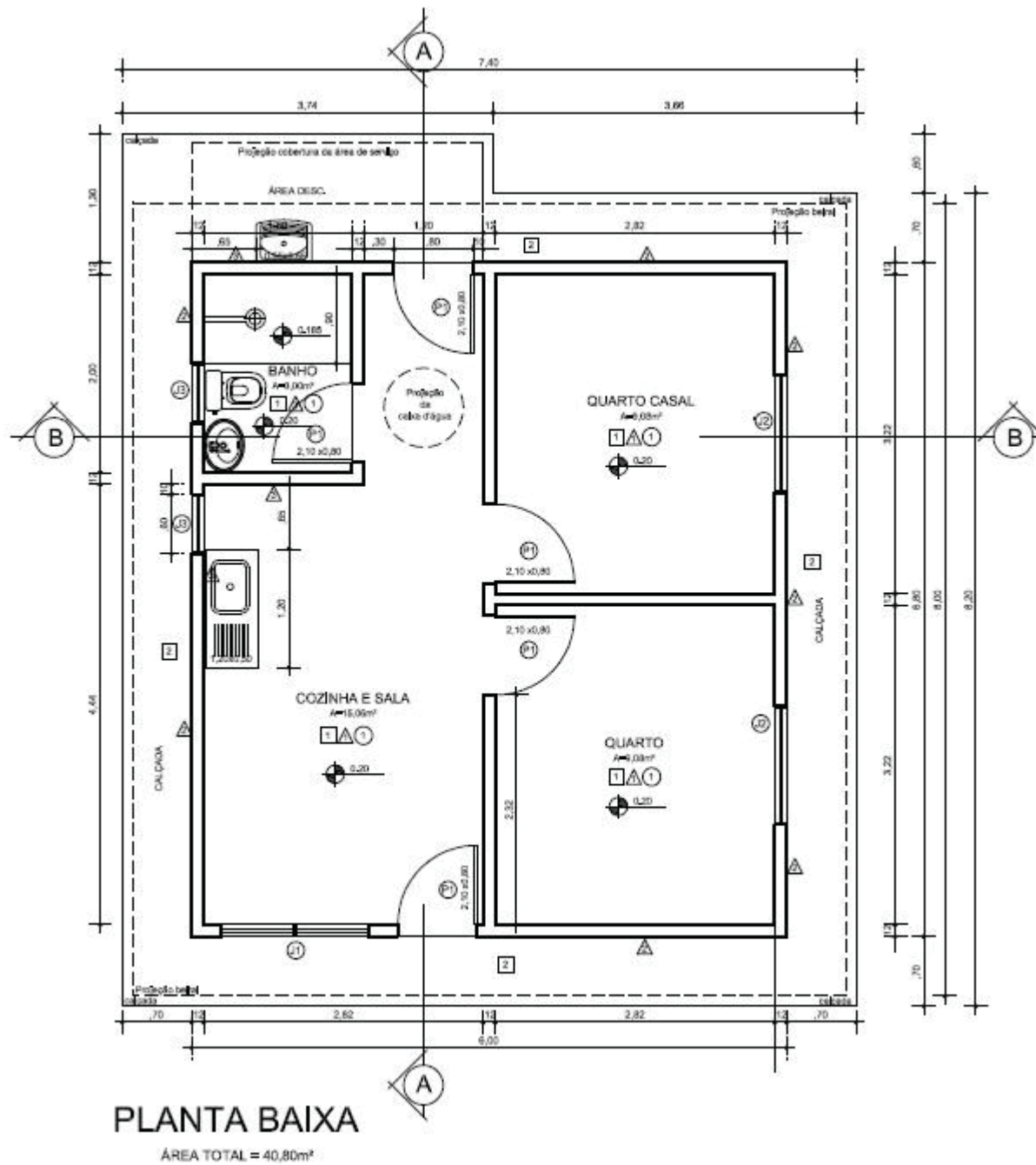
### RESULTADOS DOS EXPERIMENTOS COMPUTACIONAIS

| Nome da instância        | Nº de ativ. | Solução ótima/LB | Melhor solução GASA | Pior solução GASA | Melhor solução GA | Pior solução GA | Tempo GASA (s) | Tempo GA (s) | Melhor erro GASA | Pior erro GASA | Melhor erro GA | Pior erro GA | Iterações GASA | Iterações GA |
|--------------------------|-------------|------------------|---------------------|-------------------|-------------------|-----------------|----------------|--------------|------------------|----------------|----------------|--------------|----------------|--------------|
| Kim e Ellis Jr. (2008)   | 11          | 38               | 38                  | 38                | 38                | 40              | 45.95          | 30.51        | 0.00%            | 0.00%          | 0.00%          | 5.26%        | 15.5           | 24.8         |
| Bettemir e Sonmez (2013) | 8           | 108              | 108                 | 113               | 108               | 113             | 51.10          | 37.69        | 0.00%            | 4.63%          | 0.00%          | 4.63%        | 6.9            | 8.6          |
| Bettemir e Sonmez (2013) | 40          | 100              | 104                 | 113               | 101               | 109             | 1,220.77       | 591.30       | 4.00%            | 13.00%         | 1.00%          | 9.00%        | 21.33          | 29.9         |
| j3010_10                 | 30          | 41               | 42                  | 47                | 42                | 46              | 704,20         | 452,43       | 2,44%            | 14,63%         | 2,44%          | 12,20%       | 44             | 58           |
| j3029_7                  | 30          | 73               | 75                  | 79                | 75                | 79              | 958,86         | 503,40       | 2,74%            | 8,22%          | 2,74%          | 8,22%        | 61             | 65           |
| j3038_10                 | 30          | 60               | 60                  | 63                | 60                | 61              | 497,45         | 406,49       | 0,00%            | 5,00%          | 0,00%          | 1,67%        | 34             | 52           |
| j3030_3                  | 30          | 55               | 56                  | 62                | 56                | 62              | 1.008,29       | 502,97       | 1,82%            | 12,73%         | 1,82%          | 12,73%       | 74             | 68           |
| j305_1                   | 30          | 53               | 56                  | 61                | 54                | 62              | 595,11         | 439,58       | 5,66%            | 15,09%         | 1,89%          | 16,98%       | 41             | 60           |
| j3044_7                  | 30          | 42               | 42                  | 42                | 42                | 42              | 204,35         | 176,95       | 0,00%            | 0,00%          | 0,00%          | 0,00%        | 14             | 24           |
| j3033_2                  | 30          | 60               | 60                  | 63                | 60                | 62              | 302,12         | 277,42       | 0,00%            | 5,00%          | 0,00%          | 3,33%        | 21             | 36           |
| j304_2                   | 30          | 60               | 60                  | 60                | 60                | 60              | 129,18         | 59,94        | 0,00%            | 0,00%          | 0,00%          | 0,00%        | 8              | 7            |
| j3047_4                  | 30          | 49               | 49                  | 52                | 49                | 52              | 888,26         | 476,84       | 0,00%            | 6,12%          | 0,00%          | 6,12%        | 57             | 65           |
| j3035_10                 | 30          | 59               | 59                  | 59                | 59                | 59              | 113,58         | 60,29        | 0,00%            | 0,00%          | 0,00%          | 0,00%        | 8              | 8            |
| j3043_6                  | 30          | 58               | 58                  | 63                | 58                | 61              | 692,52         | 504,34       | 0,00%            | 8,62%          | 0,00%          | 5,17%        | 50             | 71           |
| j3016_4                  | 30          | 47               | 47                  | 47                | 47                | 47              | 64,27          | 37,13        | 0,00%            | 0,00%          | 0,00%          | 0,00%        | 4              | 4            |
| j3022_6                  | 30          | 52               | 53                  | 57                | 52                | 55              | 494,76         | 453,26       | 1,92%            | 9,62%          | 0,00%          | 5,77%        | 34             | 59           |
| j3018_4                  | 30          | 70               | 70                  | 70                | 70                | 70              | 271,01         | 175,61       | 0,00%            | 0,00%          | 0,00%          | 0,00%        | 15             | 19           |
| j3034_5                  | 30          | 63               | 63                  | 63                | 63                | 63              | 179,17         | 113,13       | 0,00%            | 0,00%          | 0,00%          | 0,00%        | 12             | 14           |
| j606_3                   | 60          | 72               | 76                  | 81                | 74                | 78              | 1.785,79       | 1.499,63     | 5,56%            | 12,50%         | 2,78%          | 8,33%        | 22             | 36           |
| j6015_4                  | 60          | 75               | 75                  | 75                | 75                | 75              | 1.454,70       | 1.241,30     | 0,00%            | 0,00%          | 0,00%          | 0,00%        | 11             | 18           |
| j6034_2                  | 60          | 68               | 72                  | 76                | 70                | 76              | 1.289,36       | 1.213,79     | 5,88%            | 11,76%         | 2,94%          | 11,76%       | 16             | 29           |
| j607_7                   | 60          | 89               | 89                  | 92                | 89                | 89              | 875,57         | 1.003,22     | 0,00%            | 3,37%          | 0,00%          | 0,00%        | 7              | 14           |
| j6026_10                 | 60          | 85               | 89                  | 92                | 88                | 95              | 1.380,75       | 1.323,78     | 4,71%            | 8,24%          | 3,53%          | 11,76%       | 19             | 35           |
| j6019_10                 | 60          | 78               | 78                  | 78                | 78                | 78              | 324,39         | 187,39       | 0,00%            | 0,00%          | 0,00%          | 0,00%        | 4              | 4            |
| j6012_7                  | 60          | 71               | 71                  | 71                | 71                | 71              | 133,03         | 68,51        | 0,00%            | 0,00%          | 0,00%          | 0,00%        | 1              | 1            |
| j608_2                   | 60          | 61               | 61                  | 61                | 61                | 61              | 996,21         | 803,89       | 0,00%            | 0,00%          | 0,00%          | 0,00%        | 7              | 11           |
| j6010_5                  | 60          | 79               | 84                  | 87                | 84                | 89              | 2.622,49       | 1.524,34     | 6,33%            | 10,13%         | 6,33%          | 12,66%       | 21             | 22           |
| j602_3                   | 60          | 78               | 78                  | 83                | 78                | 85              | 2.318,26       | 1.504,17     | 0,00%            | 6,41%          | 0,00%          | 8,97%        | 17             | 21           |
| j6042_10                 | 60          | 87               | 89                  | 97                | 91                | 96              | 2.117,10       | 1.517,86     | 2,30%            | 11,49%         | 4,60%          | 10,34%       | 32             | 45           |
| j608_3                   | 60          | 79               | 79                  | 79                | 79                | 79              | 254,76         | 130,22       | 0,00%            | 0,00%          | 0,00%          | 0,00%        | 3              | 3            |
| j6014_7                  | 60          | 69               | 75                  | 81                | 76                | 79              | 1.668,25       | 1.498,66     | 8,70%            | 17,39%         | 10,14%         | 14,49%       | 24             | 39           |

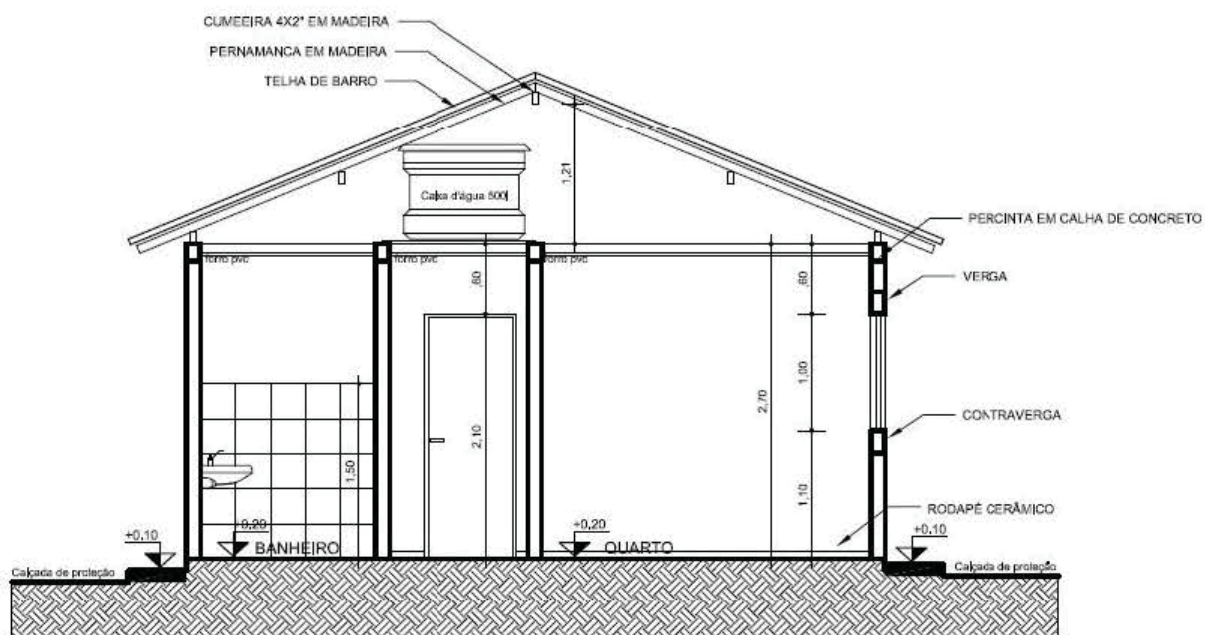
|          |    |     |     |     |     |     |          |          |        |        |        |        |    |    |
|----------|----|-----|-----|-----|-----|-----|----------|----------|--------|--------|--------|--------|----|----|
| j603_4   | 60 | 81  | 81  | 81  | 81  | 81  | 678,53   | 481,26   | 0,00%  | 0,00%  | 0,00%  | 0,00%  | 6  | 8  |
| j6024_8  | 60 | 81  | 81  | 81  | 81  | 81  | 613,29   | 340,78   | 0,00%  | 0,00%  | 0,00%  | 0,00%  | 5  | 6  |
| j9017_7  | 90 | 80  | 97  | 102 | 96  | 102 | 3.524,73 | 2.447,58 | 21,25% | 27,50% | 20,00% | 27,50% | 13 | 17 |
| j9011_4  | 90 | 64  | 67  | 70  | 64  | 70  | 2.859,22 | 2.119,48 | 4,69%  | 9,38%  | 0,00%  | 9,38%  | 11 | 15 |
| j9012_6  | 90 | 81  | 81  | 81  | 81  | 81  | 296,40   | 155,67   | 0,00%  | 0,00%  | 0,00%  | 0,00%  | 1  | 1  |
| j906_4   | 90 | 80  | 80  | 86  | 80  | 88  | 3.059,04 | 2.354,33 | 0,00%  | 7,50%  | 0,00%  | 10,00% | 11 | 15 |
| j901_4   | 90 | 86  | 104 | 111 | 103 | 111 | 3.552,95 | 2.457,52 | 20,93% | 29,07% | 19,77% | 29,07% | 13 | 17 |
| j903_9   | 90 | 61  | 61  | 62  | 61  | 62  | 2.186,44 | 1.600,88 | 0,00%  | 1,64%  | 0,00%  | 1,64%  | 8  | 11 |
| j9011_10 | 90 | 81  | 81  | 83  | 81  | 83  | 2.908,96 | 2.419,40 | 0,00%  | 2,47%  | 0,00%  | 2,47%  | 12 | 19 |
| j9013_1  | 90 | 127 | 169 | 176 | 169 | 174 | 4.305,20 | 2.487,61 | 33,07% | 38,58% | 33,07% | 37,01% | 15 | 16 |
| j902_6   | 90 | 67  | 68  | 75  | 68  | 73  | 3.373,76 | 2.362,38 | 1,49%  | 11,94% | 1,49%  | 8,96%  | 12 | 16 |
| j904_4   | 90 | 92  | 92  | 92  | 92  | 92  | 358,39   | 224,43   | 0,00%  | 0,00%  | 0,00%  | 0,00%  | 1  | 1  |
| j9010_7  | 90 | 83  | 87  | 95  | 88  | 91  | 3.661,69 | 2.370,92 | 4,82%  | 14,46% | 6,02%  | 9,64%  | 14 | 17 |
| j9015_7  | 90 | 82  | 82  | 82  | 82  | 82  | 2.312,75 | 1.344,89 | 0,00%  | 0,00%  | 0,00%  | 0,00%  | 9  | 10 |
| j9010_10 | 90 | 75  | 88  | 94  | 88  | 93  | 3.241,79 | 2.301,18 | 17,33% | 25,33% | 17,33% | 24,00% | 13 | 17 |
| j9017_3  | 90 | 89  | 105 | 114 | 104 | 109 | 3.062,92 | 2.493,83 | 17,98% | 28,09% | 16,85% | 22,47% | 10 | 16 |

## APÊNDICE C

### PROJETO USADO NA FASE DE SIMULAÇÃO

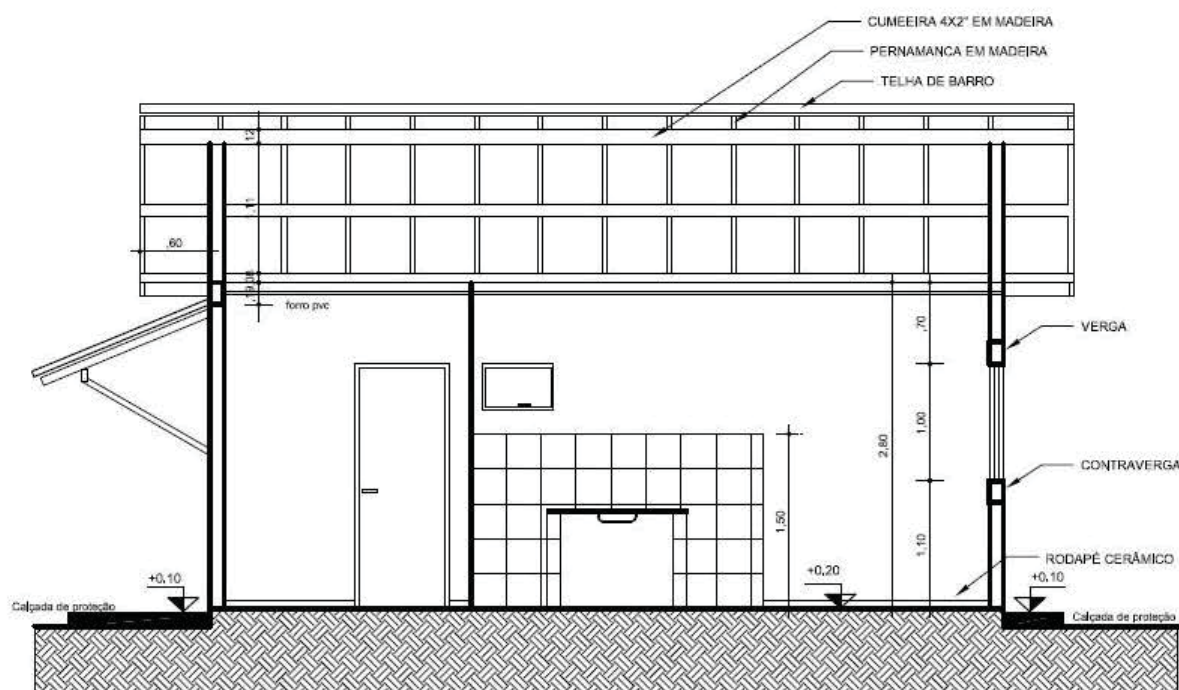






## CORTE BB'

S/ ESC.



## CORTE AA'

S/ ESC.



## APÊNDICE D

### ORGANIZAÇÃO DAS INFORMAÇÕES NAS INSTÂNCIAS DA PSPLIB

Nas figuras abaixo, é possível observar como as informações são apresentadas nos arquivos disponíveis na PSPLIB. Primeiramente são apresentadas as redes de precedência entre as atividades, onde cada uma pode estar relacionada a no máximo três atividades sucessoras.

Em seguida encontram-se as durações para cada atividade e seus requisitos de recursos. As instâncias utilizadas neste trabalho utilizam 4 tipos diferentes de recursos, sendo cada um deles utilizados em quantidades diferentes por cada atividade. Por fim, a última informação vista nos arquivos são as quantidades disponíveis para utilização de cada recurso durante toda a duração do projeto.

FIGURA 1 – RELAÇÕES DE PRECEDÊNCIA ENTRE AS ATIVIDADES

```
--
13 PROJECT INFORMATION:
14 pronr.  #jobs rel.date duedate tardcost  MPM-Time
15      1      30      0      43      0      43
16 *****
17 PRECEDENCE RELATIONS:
18 jobnr.   #modes #successors  successors
19      1         1         3      2  3  4
20      2         1         2     23 24
21      3         1         3      5  6 17
22      4         1         2      7 20
23      5         1         3     10 22 28
24      6         1         1     18
25      7         1         3      8  9 12
26      8         1         3     14 21 27
27      9         1         2     11 16
28     10         1         1     16
29     11         1         1     17
30     12         1         3     13 15 16
31     13         1         1     30
32     14         1         1     19
33     15         1         1     26
34     16         1         1     25
35     17         1         1     24
36     18         1         1     21
37     19         1         1     25
38     20         1         2     25 27
39     21         1         1     22
40     22         1         2     29 30
41     23         1         1     31
42     24         1         1     27
43     25         1         1     26
44     26         1         1     30
45     27         1         1     28
46     28         1         1     31
47     29         1         1     32
48     30         1         1     32
49     31         1         1     32
50     32         1         0
51 *****
```

Fonte: o autor, 2019

FIGURA 2 – INFORMAÇÕES DE DURAÇÃO DAS ATIVIDADES E REQUISITOS E  
DISPONIBILIDADE DE RECURSOS DO PROJETO

|    |                         |      |          |     |     |     |     |
|----|-------------------------|------|----------|-----|-----|-----|-----|
| 52 | REQUESTS/DURATIONS:     |      |          |     |     |     |     |
| 53 | jobnr.                  | mode | duration | R 1 | R 2 | R 3 | R 4 |
| 54 | -----                   |      |          |     |     |     |     |
| 55 | 1                       | 1    | 0        | 0   | 0   | 0   | 0   |
| 56 | 2                       | 1    | 1        | 0   | 0   | 0   | 5   |
| 57 | 3                       | 1    | 1        | 0   | 3   | 0   | 0   |
| 58 | 4                       | 1    | 1        | 8   | 0   | 0   | 0   |
| 59 | 5                       | 1    | 7        | 0   | 0   | 2   | 0   |
| 60 | 6                       | 1    | 6        | 0   | 0   | 0   | 3   |
| 61 | 7                       | 1    | 4        | 1   | 0   | 0   | 0   |
| 62 | 8                       | 1    | 5        | 0   | 0   | 10  | 0   |
| 63 | 9                       | 1    | 8        | 0   | 0   | 3   | 0   |
| 64 | 10                      | 1    | 7        | 0   | 0   | 0   | 1   |
| 65 | 11                      | 1    | 8        | 9   | 0   | 0   | 0   |
| 66 | 12                      | 1    | 1        | 7   | 0   | 0   | 0   |
| 67 | 13                      | 1    | 2        | 0   | 3   | 0   | 0   |
| 68 | 14                      | 1    | 3        | 0   | 0   | 0   | 6   |
| 69 | 15                      | 1    | 10       | 0   | 7   | 0   | 0   |
| 70 | 16                      | 1    | 10       | 3   | 0   | 0   | 0   |
| 71 | 17                      | 1    | 2        | 0   | 0   | 3   | 0   |
| 72 | 18                      | 1    | 10       | 0   | 0   | 4   | 0   |
| 73 | 19                      | 1    | 1        | 0   | 0   | 0   | 3   |
| 74 | 20                      | 1    | 1        | 0   | 0   | 7   | 0   |
| 75 | 21                      | 1    | 7        | 0   | 2   | 0   | 0   |
| 76 | 22                      | 1    | 9        | 0   | 0   | 0   | 10  |
| 77 | 23                      | 1    | 9        | 0   | 0   | 7   | 0   |
| 78 | 24                      | 1    | 4        | 0   | 4   | 0   | 0   |
| 79 | 25                      | 1    | 4        | 0   | 3   | 0   | 0   |
| 80 | 26                      | 1    | 1        | 0   | 0   | 4   | 0   |
| 81 | 27                      | 1    | 1        | 9   | 0   | 0   | 0   |
| 82 | 28                      | 1    | 8        | 0   | 0   | 0   | 9   |
| 83 | 29                      | 1    | 1        | 0   | 0   | 0   | 1   |
| 84 | 30                      | 1    | 2        | 0   | 8   | 0   | 0   |
| 85 | 31                      | 1    | 7        | 0   | 4   | 0   | 0   |
| 86 | 32                      | 1    | 0        | 0   | 0   | 0   | 0   |
| 87 | *****                   |      |          |     |     |     |     |
| 88 | RESOURCEAVAILABILITIES: |      |          |     |     |     |     |
| 89 | R 1                     | R 2  | R 3      | R 4 |     |     |     |
| 90 | 10                      | 8    | 13       | 12  |     |     |     |

Fonte: o autor, 2019